UNIVERSITY OF CALIFORNIA SAN DIEGO

Sparse Sampling for Appearance Acquisition

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Zexiang Xu

Committee in charge:

Professor Ravi Ramamoorthi, Chair
Professor Manmohan Chandraker
Professor David Kriegman
Professor Hao Su
Professor Zhuowen Tu

2020

The Dissertation of Zexiang Xu is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California San Diego

2020

DEDICATION

To my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to give my sincere gratitude to my advisor, Prof. Ravi Ramamoorthi, who has been enormously supportive throughout my Ph.D. journey. From Ravi, I learned the beauty of doing research and the elegancy in academic writing and presentation. His generous guidance and incredible dedication are invaluable to me, which will continue influencing my career. I also want to thank Prof. Manmohan Chandracker, Prof. Hao Su, Prof. David Kriegman and Prof. Zhuowen Tu for serving on my thesis committee and offering their time and helpful comments.

This work cannot be completed without my amazing collaborators. Thanks to Prof. Hao Su, Prof. Henrik Wann Jensen, Dr. Kalyan Sunkavalli, Dr. Sunil Hadap, Dr. Zhuwen Li, Dr. Li Erran Li, and others for their crucial contribution and constructive discussions. Thanks to Dr. Jannik Boll Nielsen, Jiyang Yu, Sai Bi, Shuo Cheng, Shilin Zhu, and others for their excellent hands-on work. I really appreciate the long-term collaboration with Adobe Research, starting from my internship with Kalyan and Sunil; Kalyan's insightful suggestions and immediate help have been constructive in many projects. My research also benefits from many other collaborative works – thanks to Prof. Manmohan Chandraker, Prof. David Kriegman, Prof. Ling-Qi Yan, Dr. Miloš Hašan, Dr. Jonathan T. Barron, Dr. Lvdi Wang, Zhengqin Li, Tiancheng Sun, Alexandr Kuznetsov, and others. I enjoyed every collaboration and I have learned so much from my collaborators. I also want to thank the entire Ravi's group and many other labmates for making the pleasant academic environment at UC San Diego for me to learn and evolve in the journey.

I'm very grateful to my Bachelor's and Master's research advisors, Prof. Qinping Zhao and Prof. Yue Qi, who provided me the early opportunity of doing research and opened the gate of computer graphics and computer vision for me. I want to thank Dr. Xin Tong, who offered me my first research internship, trained the junior me with his personal patience and strictness, and helped me along the way.

I want to thank all my friends at UC San Diego: Sai Bi, Lifan Wu, Mengting Wan, Yao Qin, Zhen Zhai, Xiaojian Chen, Tiancheng Sun, Zhengqin Li, Julaiti Alafate, Songbai Yan, Wangcheng Kang, Guo Li, etc. Thanks to all my world-wide friends: Lu Pang, Kewen Han, Zehua Huang, Weihua Li, Jiahao Guo, etc. I'd love to thank Haoxin Meng for accompanying me for so many deadlines and unforgettable moments in my Ph.D. life. I'd like to thank my parents, Xinfeng Xu and Xiuping Hu, for giving me my life; I'm extremely fortunate to have their deepest love and unconditional support. Great thanks to my entire family for their unlimited support in my life.

The work is supported in part by NSF grants 1451828, 1451830, NSF grants 1617234, 1703957, NSF grant IIS-1764078, ONR grant N000141712687, Kuaishou Technology, Adobe (including an Adobe Research Fellowship), a Powell-Bundle Fellowship, the Ronald L. Graham Chair and the UC San Diego Center for Visual Computing. We also acknowledge computing resources from the San Diego Supercomputer Center, and NSF Chase-CI 1730158.

Chapter 2 is a reformatted version of the material as it appears in “Deep Stereo using Adaptive Thin Volume Representation with Uncertainty Awareness,” Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, Hao Su. [25]. The material has been submitted and accepted to the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. The dissertation author was the co-primary investigator and author of this paper.

Chapter 3 is a reformatted version of the material as it appears in “Minimal BRDF Sampling for Two-Shot Near-Field Reflectance Acquisition,” Zexiang Xu, Jannik Boll Nielsen, Jiyang, Yu, Henrik Wann Jensen, Ravi Ramamoorthi, ACM Transactions on Graphics (TOG) 35 (6), 2016 [171]. The dissertation author was the primary investigator and author of this paper.

Chapter 4 is a reformatted version of the material as it appears in “Deep Image-Based Relighting from Optimal Sparse Samples,” Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, Ravi Ramamoorthi, ACM Transactions on Graphics (TOG) 37 (4), 2018 [172]. The dissertation author was the primary investigator and author of this paper.

Chapter 5 is a reformatted version of the material as it appears in “Deep View Synthesis from Sparse Photometric Images,” Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, Ravi Ramamoorthi, ACM Transactions on Graphics (TOG) 38 (4), 2019 [170]. The dissertation author was the primary investigator and author of this paper.

VITA

2008–2012	B.S. in Computer Science, Beihang University, China
2012–2015	M.S. in Computer Technology, Beihang University, China
2015–2020	Ph.D. in Computer Science, University of California San Diego, USA

PUBLICATIONS

Shuo Cheng*, **Zexiang Xu***, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, Hao Su. "Deep Stereo using Adaptive Thin Volume Representation with Uncertainty Awareness," to appear in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. (*Equal contribution.)

Sai Bi, **Zexiang Xu**, Kalyan Sunkavalli, David Kriegman, Ravi Ramamoorthi. "Deep 3D Capture: Geometry and Reflectance from Sparse Multi-View Images," to appear in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

Alexandr Kuznetsov, Miloš Hašan, **Zexiang Xu**, Ling-Qi Yan, Bruce Walter, Nima Khademi Kalantari, Steve Marschner, Ravi Ramamoorthi. "Learning Generative Models for Rendering Specular Microgeometry," ACM Transactions on Graphics (TOG) 38 (6), 2019.

Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, **Zexiang Xu**, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, Ravi Ramamoorthi. "Single Image Portrait Relighting," ACM Transactions on Graphics (TOG) 38 (4), 2019.

Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, Ravi Ramamoorthi. "Deep View Synthesis from Sparse Photometric Images," ACM Transactions on Graphics (TOG) 38 (4), 2019.

Zhengqin Li, **Zexiang Xu**, Ravi Ramamoorthi, Kalyan Sunkavalli, Manmohan Chandraker. "Learning to Reconstruct Shape and Spatially-Varying Reflectance from a Single Image," ACM Transactions on Graphics (TOG) 37 (6), 2018.

Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, Ravi Ramamoorthi. "Deep Image-Based Relighting from Optimal Sparse Samples," ACM Transactions on Graphics (TOG) 37 (4), 2018.

Zhengqin Li, **Zexiang Xu**, Ravi Ramamoorthi, Manmohan Chandraker. "Robust Energy Minimization for BRDF-Invariant Shape from Light Fields," the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, Ravi Ramamoorthi. "Minimal BRDF Sampling for Two-Shot Near-Field Reflectance Acquisition," ACM Transactions on Graphics (TOG) 35 (6), 2016.

Jiyang Yu, **Zexiang Xu**, Matteo Mannino, Henrik Wann Jensen, Ravi Ramamoorthi. “Sparse Sampling for Image-Based SVBRDF Acquisition,” Workshop on Material Appearance Modeling, the Eurographics Association, 2016.

Zexiang Xu, Hsiang-Tao Wu, Lvdi Wang, Changxi Zheng, Xin Tong, Yue Qi. “Dynamic Hair Capture using Spacetime Optimization,” ACM Transactions on Graphics (TOG) 33 (6), 2014.

ABSTRACT OF THE DISSERTATION

Sparse Sampling for Appearance Acquisition

by

Zexiang Xu

Doctor of Philosophy in Computer Science

University of California San Diego, 2020

Professor Ravi Ramamoorthi, Chair

Modeling the appearance of real scenes from captured images is one key problem in computer graphics and computer vision. This traditionally requires a large number of input samples (e.g. images, light-view directions, depth hypotheses, etc.) and consumes extensive computational resources. In this dissertation, we aim to make scene acquisition more efficient and practical, and we present several approaches that successfully reduce the required number of samples in various appearance acquisition problems.

We exploit techniques to explicitly reconstruct the geometry and materials in a real scene; the two components essentially determine the scene appearance. On the geometry side, we introduce a novel deep multi-view stereo technique that can reconstruct high-quality scene geometry from a sparse set of sampling depth hypotheses. We leverage uncertainty estimation in a multi-stage cascaded network, which reconstructs highly accurate and highly complete geometry with low costs in a coarse-to-fine framework. On the material side, the reflectance of a real material is traditionally measured by tens and even hundreds of captured images. We present a novel reflectance acquisition technique that can reconstruct high-fidelity real materials from only two near-field images.

Moreover, we exploit image-based acquisition techniques that bypass explicit scene reconstruction and focus on realistic image synthesis under new conditions. We first present a novel deep neural network for image-based relighting. Our network simultaneously learns optimized input lighting directions and a relighting function. Our approach can produce photo-realistic relighting results under novel environment maps from only five images captured under five optimized directional lights. We also study the problem of view synthesis for real objects under controlled lighting, which classically requires dense input views with small baselines. We propose a novel deep learning based view synthesis technique that can synthesize photo-realistic images from novel views across six widely-spaced input views. Our network leverages visibility-aware attention information to effectively aggregate multi-view appearance. We also show that our view synthesis technique can be combined with our relighting technique to achieve novel-view relighting from sparse light-view samples.

Chapter 1

Introduction

Capturing real scenes is one of the core tasks in computer graphics and computer vision, which is crucial for many applications in visual effects, 3D modeling, VR (virtual reality), AR (augmented reality), product visualization, etc. Scene acquisition is often done by capturing images of a scene and discovering various underlying models that express the appearance of the scene in the images. In general, the appearance of a scene is determined by its materials and geometry; therefore, it requires comprehensive modeling of both the materials and the geometry in a scene to fully represent the scene and synthesize images of the scene under arbitrary conditions. Explicit geometry and material reconstructions have been extensively studied; they allow for flexible control of any scene components and rendering under any conditions. Meanwhile, image based techniques have also been broadly explored; they bypass explicit scene reconstruction and focus on realistic image synthesis under novel lighting or viewpoint. However, both reconstruction and image-based techniques are highly challenging and usually require expensive computational resources and numerous (tens and even hundreds of) input images.

The central goal of this thesis is to improve the computation and acquisition efficiencies in various acquisition problems and make scene acquisition more practical. In this dissertation, we study broad topics in scene acquisition, including both explicit reconstruction and image-based acquisition. We improve the efficiency of explicit reconstruction by introducing a low-cost learning-based multi-view stereo (MVS) geometry capture approach in Chapter 2 and a two-shot material reflectance acquisition approach in Chapter 3. We also present a relighting approach in Chapter 4 and a view synthesis approach in Chapter 5, which both leverage deep learning techniques to significantly reduce the required numbers of input images in the two problems by an order of magnitude.

Multi-view stereo (MVS) aims to reconstruct fine-grained scene geometry from multi-view images. Recently, deep learning has been widely applied in MVS and outperforms traditional methods. Previous learning-based MVS methods estimate per-view depth using plane sweep volumes (PSVs); this requires densely sampled planes for high accuracy, which is impractical for high-resolution depth because of limited memory. In Chapter 2, we propose novel adaptive thin volumes (ATVs) with low computation and memory costs, and we present a novel uncertainty-aware cascaded deep network that leverages differentiable uncertainty estimation to construct multi-stage ATVs for high-resolution depth reconstruction. Our approach enables highly accurate and highly complete geometry reconstruction in a highly efficient coarse-to-fine framework.

The appearance of an opaque material is modeled by a bidirectional reflectance distribution function (BRDF), which describes how light reflects at a surface point of a material. High-quality BRDF models are crucial for realistic image synthesis. BRDF acquisition aims to recover high-fidelity BRDFs of real materials, which however traditionally requires hundreds and even thousands of input images. In Chapter 3, we develop a method to acquire the BRDF of a homogeneous flat sample from only two images, taken by a near-field perspective camera, and lit by a directional light source. Our method uses the MERL BRDF database to determine the optimal set of light-view pairs for data-driven reflectance acquisition. We demonstrate practical near-field acquisition of high-quality BRDFs from our optimal two input images. Our method significantly reduces the required number of inputs in BRDF acquisition.

We introduce techniques for efficient explicit reconstruction of geometry and materials in Chapters 2 and 3. In contrast, image-based scene acquisition aims to avoid explicit reconstruction and achieve realistic image synthesis. In Chapter 4, we study image-based relighting that focuses on rendering realistic images of real scenes from a fixed viewpoint under different lighting; this traditionally requires hundreds and even thousands of images as input. We present a novel deep learning based image-based relighting method that can synthesize scene appearance under novel illumination from only five images captured under pre-defined directional lights. We propose to jointly learn both the optimal input light directions and a deep relighting function for high-quality relighting from sparse inputs. Our method is able to reproduce complex, high-frequency lighting effects like specularities and cast shadows, and outperforms previous image-based relighting methods that require an order of magnitude more images.

Chapter 4 exploits image-based relighting from sparse images under controlled lighting. On the other hand, there has been little progress on view synthesis from a sparse set of images under controlled conditions. In Chapter 5, we present a novel approach that can synthesize novel viewpoints across a wide range of viewing directions (covering a 60° cone) from a sparse set of just six viewing directions. Previous view synthesis and image-based rendering techniques are usually restricted to much smaller baselines; at our baselines, input images have few correspondences and large occlusions, and do not work for previous methods. Our method is based on a deep convolutional network trained to directly synthesize new views from the six input views. Our network combines 3D convolutions on a plane sweep volume with a novel per-view per-depth plane attention map prediction network to effectively aggregate multi-view appearance. Our method is able to reproduce complex appearance effects like occlusions, view-dependent specularities and hard shadows. Moreover, we demonstrate that our view synthesis technique can be used to facilitate explicit reconstruction like MVS to work with sparse images; we also demonstrate that our view synthesis technique in Chapter 5 can also be combined with our relighting technique in Chapter 4 to enable changing both lighting and view.

In Chapter 6, we summarize the contributions of the thesis and discuss promising future directions in scene acquisition. We believe the efficient acquisition techniques in this thesis would inspire future appearance acquisition techniques with sparse samples in many directions, like acquisition under natural illumination, single-image appearance acquisition, multi-view geometry and reflectance reconstruction, and dynamic scene acquisition.

Chapter 2

Uncertainty-Aware Deep Stereo

2.1 Introduction

Inferring 3D scene geometry from captured images is a core problem in computer vision and graphics with applications in 3D visualization, scene understanding, robotics and autonomous driving. Multi-view stereo (MVS) aims to reconstruct dense 3D representations from multiple images with calibrated cameras. Inspired by the success of deep convolutional neural networks (CNN), several learning-based MVS methods have been presented [71, 76, 166, 63, 144]; the most recent work leverages cost volumes in a learning pipeline [177, 68], and outperforms many traditional MVS methods [45].

At the core of the recent success on MVS [177, 68] is the application of 3D CNNs on plane sweep cost volumes to effectively infer multi-view correspondence. However, such 3D CNNs involve massive memory usage for depth estimation with high accuracy and completeness. In particular, for a large scene, high accuracy requires sampling a large number of sweeping planes and high completeness requires reconstructing high-resolution depth maps. In general, given limited memory, there is an undesired trade-off between accuracy (more planes) and completeness (more pixels) in previous work [177, 68].

Our goal is to achieve *highly accurate and highly complete reconstruction* with *low memory and computation consumption* at the same time. To do so, we propose a novel learning-based uncertainty-aware multi-view stereo framework, which utilizes multiple small volumes,

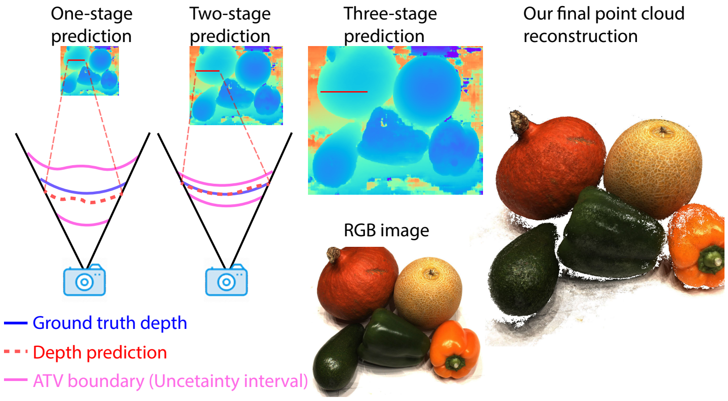


Figure 2.1. Our UCS-Net leverages adaptive thin volumes (ATVs) to progressively reconstruct a highly accurate high-resolution depth map through multiple stages. We show the input RGB image, depth predictions with increasing sizes from three stages, and our final point cloud reconstruction obtained by fusing multiple depth maps. We also illustrate a local slice (red) from our depth prediction with the corresponding ATV boundaries that reflect pixel-wise uncertainty intervals. Our ATVs become thinner after a stage with reduced uncertainty, which enables higher accuracy.

instead of a large standard plane sweep volume, to progressively regress high-quality depth in a coarse-to-fine fashion. A key in our method is that we propose novel adaptive thin volumes (ATVs, see Fig. 2.1) to achieve efficient spatial partitioning.

Specifically, we propose a novel cascaded network with three stages (see Fig. 2.2): each stage of the cascade predicts a depth map with a different size; each following stage constructs an ATV to refine the predicted depth from the previous stage with higher pixel resolution and finer depth partitioning. The first stage uses a small standard plane sweep volume with low image resolution and relatively sparse depth planes – 64 planes that are fewer than the number of planes (256 or 512) in previous work [177, 178]; the following two stages use ATVs with higher image resolutions and significantly fewer depth planes – only 32 and 8 planes. While consisting of a very small number of planes, our ATVs are constructed within *learned local depth ranges*, which enables *efficient and fine-grained spatial partitioning* for accurate and complete depth reconstruction.

This is made possible by the novel uncertainty-aware construction of an ATV. In particular, we leverage the variances of the predicted per-pixel depth probabilities, and infer the uncertainty intervals (as shown in Fig. 2.1 and Fig. 2.3) by calculating variance-based confidence intervals of the per-pixel probability distributions for the ATV construction. Specifically, we apply the previously predicted depth map as a central curved plane, and construct an ATV around the central plane within local per-pixel uncertainty intervals. In this way, we explicitly express the uncertainty of the depth prediction at one stage, and embed this knowledge into the input volume for the next stage.

Our variance-based uncertainty estimation is differentiable and we train our UCSNet from end to end with depth supervision for the predicted depths from all three stages. Our network can thus learn to optimize the estimated uncertainty intervals, to make sure that an ATV is constructed with proper depth coverage that is both large enough – to try to cover ground truth depth – and small enough – to enable accurate reconstruction for the following stages. Overall, our multi-stage framework can progressively sub-divide the local space at a finer scale in a reasonable way, which leads to high-quality depth reconstruction. We demonstrate that our novel UCS-Net outperforms the state-of-the-art learning-based MVS methods on various datasets.

2.2 Related Work

Multi-view stereo is a long-studied vision problem with many traditional approaches [134, 118, 83, 82, 75, 38, 30, 45, 131]. Our learning-based framework leverages the novel spatial representation, ATV to reconstruct high-quality depth for fine-grain scene reconstruction. In this work, we mainly discuss spatial representation for 3D reconstruction and deep learning based multi-view stereo.

Spatial Representation for 3D Reconstruction. Existing methods can be categorized based on learned 3D representations. Volumetric based approaches partition the space into a regular 3D volume with millions of small voxels [71, 76, 166, 167, 179, 124], and the network predicts if a voxel is on the surface or not. Ray tracing can be incorporated into this voxelized structure [148, 114, 150]. The main drawback of these methods is computation and memory

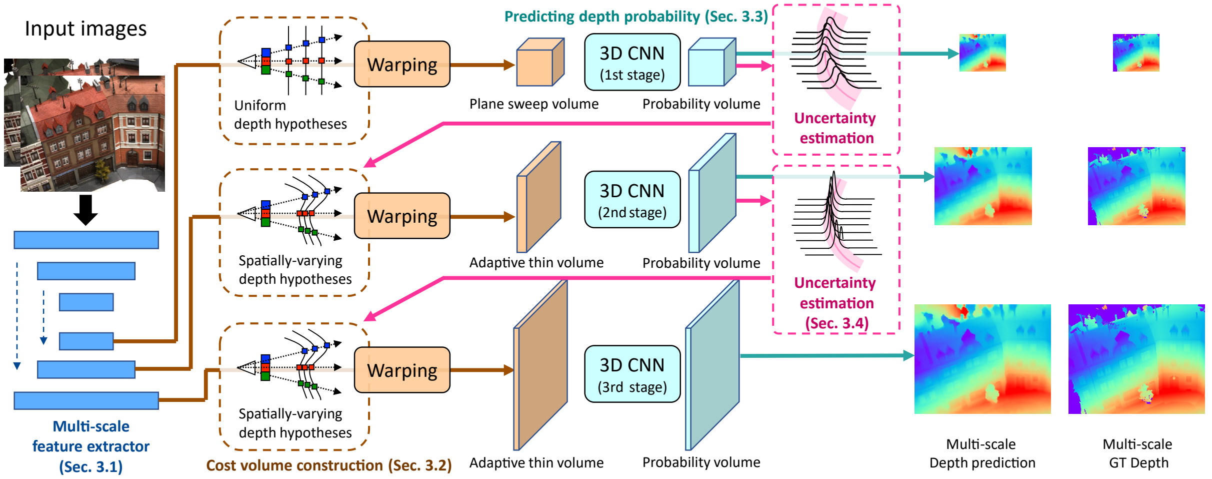


Figure 2.2. Overview of our UCS-Net. Our UCS-Net leverages multi-scale cost volumes to achieve coarse-to-fine depth prediction with three cascade stages. The cost volumes are constructed using multi-scale deep image features from a multi-scale feature extractor. The last two stages utilize the uncertainty of the previous depth prediction to build adaptive thin volumes (ATVs) for depth reconstruction at a finer scale. We mark different parts of the network in different colors. Please refer to Sec 2.3 and the corresponding subsections for more details.

inefficiency, given that most voxels are not on the surface. Researchers have also tried to reconstruct point clouds [69, 45, 88, 155, 92, 2], however the high dimensionality of a point cloud often results in noisy outliers since a point cloud does not efficiently encode connectivity between points. Some recent works utilize single or multiple images to reconstruct a point cloud given strong shape priors [39, 69, 92], which cannot be directly extended to large-scale scene reconstruction. Recent work also tried to directly reconstruct surface meshes [84, 74, 156, 59, 137, 78], deformable shapes [73, 74], and some learned implicit distance functions [27, 125, 103, 24]. These reconstructed surfaces often look smoother than point-cloud-based approaches, but often lack high-frequency details. A depth map represents dense 3D information that is perfectly aligned with a reference view; depth reconstruction has been demonstrated in many previous works on reconstruction with both single view [34, 151, 49, 53, 182] and multiple views [15, 146, 57, 46, 131, 176]. Some of them leverage normal information as well [46, 47]. In this chapter, we present ATV, a novel spatial representation for depth estimation; we use two ATVs to progressively partition local space, which is the key to achieve coarse-to-fine reconstruction.

Deep Multi-View Stereo (MVS). The traditional MVS pipeline mainly relies on photo-consistency constraints to infer the underlying 3D geometry, but usually performs poorly on texture-less or occluded areas, or under complex lighting environments. To overcome such limitations, many deep learning-based MVS methods have emerged in the last two years, including regression-based approaches [177, 68], classification-based approaches [63] and approaches based on recurrent- or iterative- style architectures [178, 180, 22] and many other approaches [80, 114, 8, 136]. Most of these methods build a single cost volume with uniformly sampled depth hypotheses by projecting 2D image features into 3D space, and then use a stack of either 2D or 3D CNNs to infer the final depth [177, 40, 170]. However, a single cost volume often requires a large number of depth planes to achieve enough reconstruction accuracy, and it is difficult to reconstruct high-resolution depth, limited by the memory bottleneck. R-MVSNet [178] leverages recurrent networks to sequentially build a cost volume with a high depth-wise sampling rate (512 planes). In contrast, we apply an adaptive sampling strategy with ATVs, which enables more efficient spatial partitioning with a higher depth-wise sampling rate using fewer depth planes (104 planes in total, see Tab. 2.5), and our method achieves significantly better reconstruction than R-MVSNet (see Tab. 2.3 and Tab. 2.4). On the other hand, Point-MVSNet [22] densifies a coarse reconstruction within a predefined local spatial range for better reconstruction with learning-based refinement. We propose to refine depth in a learned local space with adaptive thin volumes to obtain accurate high-resolution depth, which leads to better reconstruction than Point-MVSNet and other state-of-the-art methods (see Tab. 2.3 and Tab. 2.4).

2.3 Method

Some recent works aim to improve learning-based MVS methods. Recurrent networks [178] have been utilized to achieve fine depth-wise partitioning for high accuracy; a PointNet-based method [22] is also presented to densify the reconstruction for high completeness. Our goal is to reconstruct high-quality 3D geometry with both high accuracy and high completeness. To this end, we propose a novel uncertainty-aware cascaded network (UCS-Net) to reconstruct highly accurate per-view depth with high resolution.

Given a reference image $\mathbf{I}_1$ and $N-1$ source images $\{\mathbf{I}_i\}_{i=2}^{N}$, our UCS-Net progressively regresses a fine-grained depth map at the same resolution as the reference image. We show the architecture of the UCS-Net in Fig. 2.2. Our UCS-Net first leverages a 2D CNN to extract multi-scale deep image features at three resolutions (Sec. 2.3.1). Our depth prediction is achieved through three stages, which leverage multi-scale image features to predict multi-resolution depth maps. In these stages, we construct multi-scale cost volumes (Sec. 2.3.2), where each volume is a plane sweep volume or an adaptive thin volume (ATV). We then apply 3D CNNs to process the cost volumes to predict per-pixel depth probability distributions, and a depth map is reconstructed from the expectations of the distributions (Sec. 2.3.3). To achieve efficient spatial partitioning, we utilize the uncertainty of the depth prediction to construct ATVs as cost volumes for the last two stages (Sec. 2.3.4). Our multi-stage network effectively reconstructs depth in a coarse-to-fine fashion (Sec. 2.3.5).

2.3.1 Multi-scale Feature Extractor

Previous methods use downsampling layers [177, 178] or a UNet [170] to extract deep features and build a plane sweep volume at a single resolution. To reconstruct high-resolution depth, we introduce a multi-scale feature extractor, which enables constructing multiple cost volumes at different scales for multi-resolution depth prediction. As schematically shown in Fig. 2.2, our feature extractor is a small 2D UNet [129], which has an encoder and a decoder with skip connections. The encoder consists of a set of convolutional layers followed by BN (batch normalization) and ReLu activation layers; we use stride = 2 convolutions to downsample the original image size twice. The decoder upsamples the feature maps, convolves the upsampled features and the concatenated features from skip links, and also applies BN and Relu layers. The details of the feature extractor are shown in Tab. 2.1. Given each input image $\mathbf{I}_i$, the feature

Table 2.1. We show the detailed networdk architecture of our multi-scale feature extractor, which is a 2D CNN. We highlight the three layers that output the three-scale features.

Layer	Stride	Kernel	Channel	Input
conv_unit0_0	1x1	3x3	3->8	rgb
conv_unit0_1	1x1	3x3	8->8	conv_unit0_0
conv_unit1_0	2x2	5x5	8->16	conv_unit0_1
conv_unit1_1	1x1	3x3	16->16	conv_unit1_0
conv_unit1_2	1x1	3x3	16->16	conv_unit1_1
conv_unit2_0	2x2	5x5	16->32	conv_unit1_2
conv_unit2_1	1x1	3x3	32->32	conv_unit2_0
conv_unit2_2	1x1	3x3	32->32	conv_unit2_1
conv_out1	1x1	1x1	32->32	conv_unit2_2
deconv_unit1_0	2x2	3x3	32->16	conv_unit2_2
concat1	-	-	-	deconv_unit1_0, conv_unit1_2
conv_unit3_0	1x1	3x3	32->16	concat1
conv_out2	1x1	1x1	16->16	conv_unit3_0
deconv_unit2_0	2x2	3x3	16->8	conv_unit3_0
concat2	-	-	-	deconv_unit2_0, conv_unit0_1
conv_unit4_0	1x1	3x3	16->8	concat2
conv_out3	1x1	1x1	8->8	conv_unit4_0

extractor provides three scale feature maps, $F_{i,1}$, $F_{i,2}$, $F_{i,3}$, from the decoder for the following cost volume construction. We represent the original image size as $W \times H$, where W and H denote the image width and height; correspondingly, $F_{i,1}$, $F_{i,2}$ and $F_{i,3}$ have resolutions of $\frac{W}{4} \times \frac{H}{4}$, $\frac{W}{2} \times \frac{H}{2}$ and $W \times H$, and their numbers of channels are 32, 16 and 8 respectively. Our multi-scale feature extractor allows for the high-resolution features to properly incorporate the information at lower resolutions through the learned upsampling process; thus in the multi-stage depth prediction, each stage is aware of the meaningful feature knowledge used in previous stages, which leads to reasonable high-frequency feature extraction.

2.3.2 Cost Volume Construction

We construct multiple cost volumes at multiple scales by warping the extracted feature maps, $F_{i,1}$, $F_{i,2}$, $F_{i,3}$ from source views to a reference view. Similar to previous work, this process is achieved through differentiable unprojection and projection. In particular, given camera intrinsic and extrinsic matrices $\{K_i, T_i\}$ for each view i, the 4×4 warping matrix at depth d at the reference view is given by:

$$H_i(d) = K_i T_i T_1^{-1} K_1^{-1}. \tag{2.1}$$

In particular, when warping to a pixel in the reference image $\mathbf{I}_1$ at location (x, y) and depth d, $H_i(d)$ multiplies the homogeneous vector $(xd, yd, d, 1)$ to finds its corresponding pixel location in each $\mathbf{I}_i$ in homogeneous coordinates.

Each cost volume consists of multiple planes; we use $\mathbf{L}_{k,j}$ to denote the depth hypothesis of the jth plane at the kth stage, and $\mathbf{L}_{k,j}(x)$ represents its value at pixel x. At stage k, once we warp per-view feature maps $F_{i,k}$ at all depth planes with corresponding hypotheses $\mathbf{L}_{k,j}$, we calculate the variance of the warped feature maps across views at each plane to construct a cost volume. We use D_k to represent the number of planes for stage k. For the first stage, we build a standard plane sweep volume, whose depth hypotheses are of constant values, i.e. $\mathbf{L}_{1,j}(x) = d_j$. We uniformly sample $\{d_j\}_{j=1}^{D_1}$ from a pre-defined depth interval $[d_{min}, d_{max}]$ to construct the volume, in which each plane is constructed using $H_i(d_j)$ to warp multi-view images. For the second and third stages, we build novel adaptive thin volumes, whose depth hypotheses have spatially-varying depth values according to pixel-wise uncertainty estimates of the previous depth prediction. In this case, we calculate per-pixel per-plane warping matrices by setting $d = \mathbf{L}_{k,j}(x)$ in Eqn. 2.1 to warp images and construct cost volumes. Please refer to Sec. 2.3.4 for uncertainty estimation.

2.3.3 Depth Prediction and Probability Distribution

At each stage, we apply a 3D CNN to process the cost volume, infer multi-view correspondence and predict depth probability distributions. In particular, we use a 3D UNet similar to [177], which has multiple downsampling and upsampling 3D convolutional layers to reason

Table 2.2. We show the detailed network architecture of our 3D CNN for cost volume processing.

Layer	Stride	Kernel	Channel	Input
conv_unit0	1x1x1	3x3x3	8->8	cost volume
conv_unit1	2x2x2	3x3x3	8->16	conv_unit0
conv_unit2	1x1x1	3x3x3	16->16	conv_unit1
conv_unit3	2x2x2	3x3x3	16->32	conv_unit2
conv_unit4	1x1x1	3x3x3	32->32	conv_unit3
conv_unit5	2x2x2	3x3x3	32->64	conv_unit4
conv_unit6	1x1x1	3x3x3	64->64	conv_unit5
deconv_unit7	2x2x2	3x3x3	64->32	conv_unit6
deconv_unit8	2x2x2	3x3x3	32->16	conv_unit4 + deconv_unit7
deconv_unit9	2x2x2	3x3x3	16->8	conv_unit2 + deconv_unit8
conv_out	1x1x1	3x3x3	8->1	conv_unit0 + deconv_unit9

about scene geometry at multiple scales. We apply depth-wise softmax at the end of the 3D CNNs to predict per-pixel depth probabilities. Our three stages use the same network architecture without sharing weights, so that each stage learns to process its information at a different scale. Please refer to Tab. 2.2 for the details of our 3D CNN architecture.

The 3D CNN at each stage predicts a depth probability volume that consists of D_k depth probability maps $\mathbf{P}_{k,j}$ associated with the depth hypotheses $\mathbf{L}_{k,j}$. $\mathbf{P}_{k,j}$ expresses per-pixel depth probability distributions, where $\mathbf{P}_{k,j}(x)$ represents how probable the depth at pixel x is $\mathbf{L}_{k,j}(x)$. A depth map $\hat{\mathbf{L}}_k$ at stage k is reconstructed by weighted sum:

$$\hat{\mathbf{L}}_k(x) = \sum_{j=1}^{D_k} \mathbf{L}_{k,j}(x) \cdot \mathbf{P}_{k,j}(x). \tag{2.2}$$

2.3.4 Uncertainty Estimation and ATV

The key for our framework is to progressively sub-partition the local space and refine the depth prediction with increasing resolution and accuracy. To do so, we construct novel ATVs for the last two stages, which have curved sweeping planes with spatially-varying depth hypotheses (as illustrated in Fig. 2.1 and Fig. 2.2), based on uncertainty inference of the predicted depth in its previous stage.

Given a set of depth probability maps, previous work only utilizes the expectation of the per-pixel distributions (using Eqn. (2.2)) to determine an estimated depth map. For the first time, we leverage the variance of the distribution for uncertainty estimation, and construct ATVs using the uncertainty. In particular, the variance $\hat{\mathbf{V}}_k(x)$ of the probability distribution at pixel x and stage k is calculated as:

$$\hat{\mathbf{V}}_k(x) = \sum_{j=1}^{D_k} \mathbf{P}_{k,j}(x) \cdot (\mathbf{L}_{k,j}(x) - \hat{\mathbf{L}}_k(x))^2, \tag{2.3}$$

and the corresponding standard deviation is $\hat{\sigma}_k(x) = \sqrt{\hat{\mathbf{V}}_k}$. Given the depth prediction $\hat{\mathbf{L}}_k(x)$ and its variance $\hat{\sigma}_k(x)^2$ at pixel x, we propose to use a variance-based confidence interval to measure the uncertainty of the prediction:

$$\mathbf{C}_k(x) = [\hat{\mathbf{L}}_k(x) - \lambda\hat{\sigma}_k(x), \hat{\mathbf{L}}_k(x) + \lambda\hat{\sigma}_k(x)], \tag{2.4}$$

where λ is a scalar parameter that determines how large the confidence interval is. For each pixel x, we uniformly sample D_{k+1} depth values from $\mathbf{C}_k(x)$ of the kth stage, to get its depth values $\mathbf{L}_{k+1,1}(x)$, $\mathbf{L}_{k+1,2}(x)$,...,$\mathbf{L}_{k+1,D_{k+1}}(x)$ of the depth planes for stage $(k+1)$. In this way, we construct D_{k+1} spatially-varying depth hypotheses $\mathbf{L}_{k+1,j}$, which form the ATV for stage $(k+1)$.

The estimated $\mathbf{C}_k(x)$ expresses the uncertainty interval of the prediction $\hat{\mathbf{L}}_k(x)$, which determines the physical thickness of an ATV at each pixel. In Fig. 2.3, we show two actual examples with two pixels and their estimated uncertainty intervals $\mathbf{C}_k(x)$ around the predictions (red dash line). The $\mathbf{C}_k$ essentially depicts a probabilistic local space around the ground truth surface, and the ground truth depth is located in the uncertainty interval with a very high confidence. Note that, our variance-based uncertainty estimation is differentiable, which enables our UCS-Net to learn to adjust the probability prediction at each stage to achieve optimized

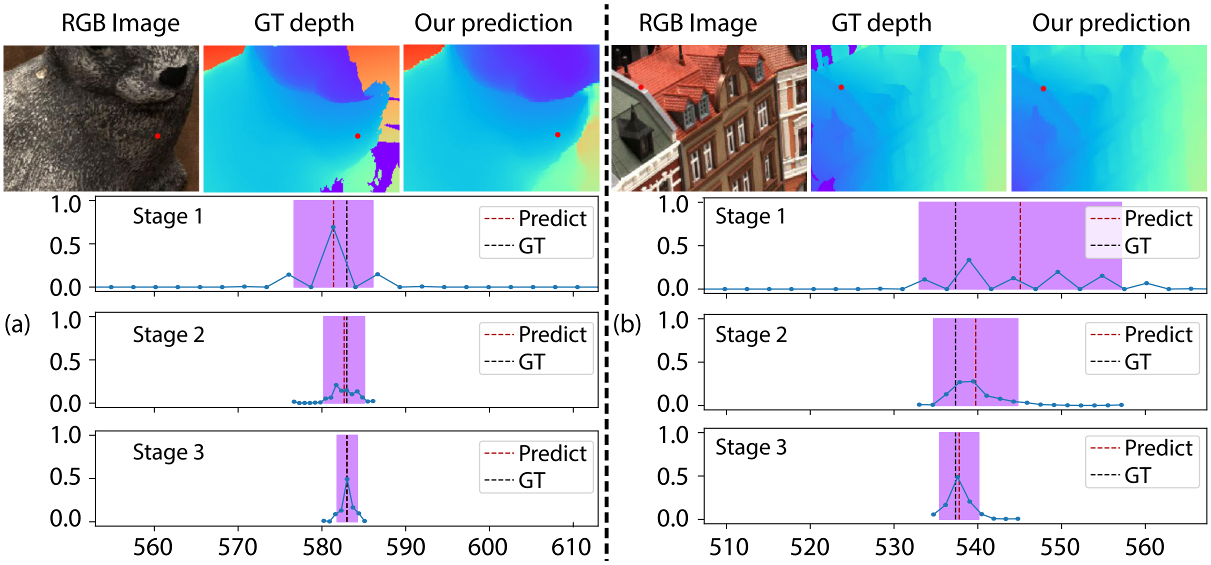


Figure 2.3. We illustrate detailed depth and uncertainty estimation of two examples. On the top, we show the RGB image crops, predicted depth and ground truth depth. On the bottom, we show the details of of two pixels (red points in the images) with predicted depth probabilities (connected blue dots) , depth prediction (red dash line), the ground truth depth (black dash line) and uncertainty intervals (purple) in the three stages.

intervals and corresponding ATVs for following stages in an end-to-end training process. As a result, the spatially varying depth hypotheses in ATVs naturally adapt to the uncertainty of depth predictions, which leads to highly efficient spatial partitioning.

2.3.5 Coarse-to-Fine Prediction

Our UCS-Net leverages three stages to reconstruct depth at multiple scales from coarse to fine, which generally supports different numbers (D_k) of planes in each stage. In practice, we use $D_1 = 64$, $D_2 = 32$ and $D_3 = 8$ to construct a plane sweep volume and two ATVs with sizes of $\frac{W}{4} \times \frac{H}{4} \times 64$, $\frac{W}{2} \times \frac{H}{2} \times 32$ and $H \times W \times 8$ to estimate depth at corresponding resolutions. While our two ATVs have small numbers (32 and 8) of depth planes, they in fact partition local depth ranges at finer scales than the first stage volume; this is achieved by our novel uncertainty-aware volume construction process which adaptively controls local depth intervals. This efficient usage of a small number of depth planes enables the last two stages to deal with higher pixel-wise resolutions given the limited memory, which makes fine-grained depth reconstruction possible.

Our novel ATV effectively expresses the locality and uncertainty in the depth prediction, which enables state-of-the-art depth reconstruction results with high accuracy and high completeness through a coarse-to-fine framework.

2.3.6 Training Details

Training set. We train our network on the DTU dataset [1]. We split the dataset into training, validate and testing set, and create ground truth depth similar to [177]. In particular, we apply Poisson reconstruction [79] on the point clouds in DTU, and render the surface at the captured views with three resolutions, $\frac{W}{4} \times \frac{H}{4}$, $\frac{W}{2} \times \frac{H}{2}$ and the original $W \times H$. In particular, we use $W \times H = 640 \times 512$ for training.

Loss function. Our UCS-Net predicts depth at three resolutions; we apply $L1$ loss on depth prediction at each resolution with the rendered ground truth at the same resolution. Our final loss is the combination of the three $L1$ losses.

Training policy. We train our full three-stage network from end to end for 60 epochs. We use Adam optimizer with an initial learning rate of 0.0016. We use 8 NVIDIA GTX 1080Ti GPUs to train the network with a batch size of 16 (mini-batch size of 2 per GPU).

2.4 Experiments

We now evaluate our UCS-Net. We do benchmarking on the DTU [1] and Tanks and Temple datasets [81]. We then justify the effectiveness of the designs of our network, in terms of uncertainty estimation and multi-stage prediction.

Evaluation on the DTU dataset [1]. We evaluate our method on the DTU testing set. To reconstruct the final point cloud, we follow [46] to fuse the depth from multiple views; we use this fusion method for all our experiments. For fair comparisons, we use the same view selection, image size and initial depth range as in [177] with $N = 5$, $W = 1600$, $H = 1184$, $d_{\min} = 425mm$ and $d_{\max} = 933.8mm$; similar settings are also used in other learning-based MVS

Table 2.3. Quantitative results of accuracy, completeness and overall on the DTU testing set. Numbers represent distances in millimeters and smaller means better.

Method	Acc.	Comp.	Overall
Camp [15]	0.835	0.554	0.695
Furu [45]	0.613	0.941	0.777
Tola [146]	0.342	1.190	0.766
Gipuma [46]	**0.283**	0.873	0.578
SurfaceNet [137]	0.450	1.040	0.745
MVSNet [177]	0.396	0.527	0.462
R-MVSNet [178]	0.383	0.452	0.417
Point-MVSNet [22]	0.342	0.411	<u>0.376</u>
Our 1st stage	0.548	0.529	0.539
Our 2nd stage	0.401	<u>0.397</u>	0.399
Our full model	<u>0.338</u>	**0.349**	**0.344**

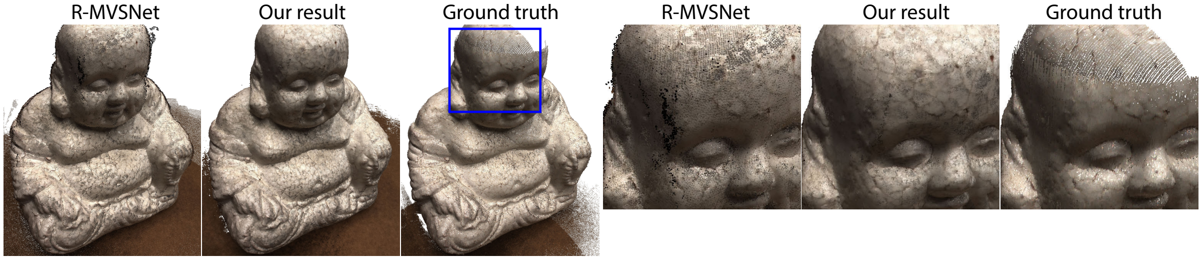


Figure 2.4. Comparisons with R-MVSNet on an example in the DTU dataset. We show rendered images of the point clouds of our method, R-MVSNet and the ground truth. In this example, the ground truth from scanning is incomplete. We also show insets for detailed comparisons marked as a blue box in the ground truth. Note that our result is smoother and has fewer outliers than R-MVSNet's result.

methods [22, 178]. We use a NVIDIA GTX 1080 Ti GPU to run the evaluation.

We compare the accuracy and the completeness of the final reconstructions using the distance metric in [1]. We compare against both traditional methods and learning-based methods, and the average quantitative results are shown in Tab. 2.3. While Gipuma [46] (a traditional method) achieves the best accuracy among all methods, our method has significantly better completeness and overall scores. Besides, our method outperforms all state-of-the-art baseline methods in terms of both accuracy and completeness. Note that with the same input, MVSNet and R-MVSNet predict depth maps with a size of only $\frac{W}{4} \times \frac{H}{4}$; our final depth maps are estimated at the original image size, which are of much higher resolution and lead to significantly better completeness. Meanwhile, such high completeness is obtained without losing accuracy; our accuracy is also significantly better thanks to our uncertainty-aware progressive reconstruction. Point-MVSNet [22] densifies low-resolution depth within a predefined local depth range, which also reconstructs depth at the original image resolution; in contrast, our UCS-Net leverages learned adaptive local depth ranges and achieves better accuracy and completeness.

We also show results from our intermediate low-resolution depth of the first and the second stages in Tab. 2.3. Note that, because of sparser depth planes, our first-stage results (64 planes) are worse than MVSNet (256 planes) and R-MVSNet (512 planes) that reconstruct depth at the same low resolution. Nevertheless, our novel uncertainty-aware network introduces highly efficient spatial partitioning with ATVs in the following stages, so that our intermediate second-stage reconstruction is already much better than the two previous methods, and our third stage further improves the quality and achieves the best reconstruction.

We show qualitative comparisons between our method and R-MVSNet [178] in Fig. 2.4, in which we use the released point cloud reconstruction on R-MVSNet's website for the comparison. While both methods achieve comparable completeness in this example, it is very hard for R-MVSNet to achieve high accuracy at the same time, which introduces obvious outliers and noise on the surface. In contrast, our method is able to obtain high completeness and high accuracy simultaneously as reflected by the smooth complete geometry in the image.

Evaluation on Tanks and Temple dataset [81]. We now evaluate the generalization of our model by testing our network trained with the DTU dataset on complex outdoor scenes in the Tanks and Temple intermediate dataset. We use $N = 5$ and $W \times H = 1920 \times 1056$ for this experiment. Our method outperforms most published methods, and to the best of our knowledge, when comparing with all published learning-based methods, we achieve the best average F-score (54.83) as shown in Tab. 2.4. In particular, our method obtains higher F-scores than MVSNet [177] and Point-MVSNet [22] in all nine testing scenes. Dense-R-MVSNet leverages a

Table 2.4. Quantitative results of F-scores (higher means better) on Tanks and Temples. From top to bottom, we compare the results from MVSNet[177], R-MVSNet[178], Dense-R-MVSNet[178], Point-MVSNet[22] and our full model.

Method	Mean	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train
[177]	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69
[178]	48.40	69.96	46.65	32.59	42.95	51.88	48.80	52.00	42.38
Dense [178]	50.55	73.01	**54.46**	**43.42**	43.88	46.80	46.69	50.87	45.25
[22]	48.27	61.79	41.15	34.20	50.79	51.97	50.85	52.38	43.06
Ours	**54.83**	**76.09**	53.16	43.03	**54.00**	**55.60**	**51.49**	**57.38**	**47.89**

well-designed post-processing method and achieves slightly better performance than ours on two of the scenes, whereas our work is focused on high-quality per-view depth reconstruction, and we use a traditional fusion technique for post-processing. Nonetheless, thanks to our high-quality depth, our method still outperforms Dense-R-MVSNet on most of the testing scenes and achieves the best overall performance.

Evaluation of uncertainty estimation. One key design of our UCS-Net is leveraging differentiable uncertainty estimation for the ATV construction. We now evaluate our uncertainty estimation on the DTU validate set. In Tab. 2.5, we show the average length of our estimated uncertainty intervals, the corresponding average sampling distances between planes, and the ratio of the pixels whose estimated uncertainty intervals cover the ground truth depth in the ATVs; we also show the corresponding values of the standard plane sweep volume (PSV) used in the first stage, which has an interval length of $d_{\max} - d_{\min} = 508.8mm$ and covers the ground truth depth with certainty. Additionally, we show distributions of the lengths of our uncertainty intervals in Appendix. A, which illustrates the details of the lengths in Tab. 2.5.

We can see that our method is able to construct efficient ATVs that cover very local depth ranges. The first ATV significantly reduces the initial depth range from 508.8mm to only 13.88mm in average, and the second ATV further reduces it to only 3.83mm. Our ATV enables efficient depth sampling in an adaptive way, and obtains about 0.48mm sampling distance with only 32 or 8 depth planes. Note that, MVSNet and R-MVSNet sample the same large depth range (508.8mm) in a uniform way with a large number of planes (256 and 512); yet, the uniform

Table 2.5. Evaluation of uncertainty estimation. The PSV is the first-stage plane sweep volume; the 1st ATV is constructed after the first stage and used in the second stage; the 2nd ATV is used in the third stage. We show the percentages of uncertainty intervals that cover the ground truth depth. We also show the average length of the intervals, the number of depth planes and the unit sampling distance.

	Ratio	Interval	D_k	Unit
PSV	100%	508.8mm	64	7.95mm
1st ATV	94.72%	13.88mm	32	0.43mm
2st ATV	85.22%	3.83mm	8	0.48mm

sampling merely obtains volumes with sampling distances of 1.99mm and 0.99mm along depth. In contrast, our UCS-Net achieves a higher actual depth-wise sampling rate with a small number of planes; this allows for the focus of the cost volumes to be changed from sampling the depth to sampling the image plane with dense pixels in ATVs given the limited memory, which enables high-resolution depth reconstruction.

Our adaptive thin volumes achieve high ratios (94.72% and 85.22%) of covering the ground truth depth in the validate set, as shown in Tab. 2.5; this justifies that our estimated uncertainty intervals are of high confidence. Our variance-based uncertainty estimation is equivalent to approximating a depth probability distribution as a Gaussian distribution and then computing its confidence interval with a specified scale on its standard deviation as in Eqn. 2.4.

We note that our variance-based uncertainty estimation is not only valid for single-mode Gaussian-like distributions as in Fig. 2.3.a, but also valid for many multi-mode cases as in Fig. 2.3.b, which shows a challenging example near object boundary. In Fig. 2.3.b, the predicted first-stage depth distribution has multiple modes; yet, it correspondingly has large variance and a large enough uncertainty interval. Our network predicts reasonable uncertainty intervals that are able to cover the ground truth depth in most cases, which allows for increasingly accurate reconstruction in the following stages at finer local spatial scales. This is made possible by the differentiable uncertainty estimation and the end-to-end training process, from which the network learns to control per-stage probability estimation to obtain proper uncertainty intervals for ATV construction. Because of this, we observe that our network is not very sensitive to different λ,

Table 2.6. Ablation study on the DTU testing set with different stages and upsampling scales (a scale of 1 represents the original result at the stage). The quantitative results represent average distances in mm (lower is better).

Stage	Scale	Size	Acc.	Comp.	Overall
1	×1	400x296	0.548	0.529	0.539
1	×2	800x592	0.411	0.535	0.473
2	×1	800x592	0.401	0.397	0.399
2	×2	1600x1184	0.342	0.386	0.364
3	×1	1600x1184	0.338	0.349	0.344

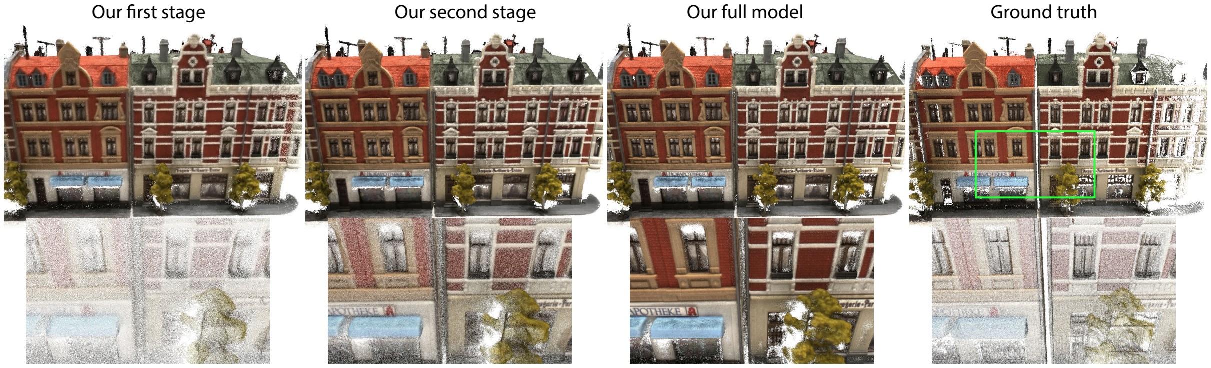


Figure 2.5. Qualitative comparisons between multi-stage point clouds and the ground truth point cloud on a scene in the DTU validate set. We show zoom-out (top) and zoom-in (bottom) rendered point clouds; the corresponding zoom-in region is marked in the ground truth as a green box. Our UCS-Net achieves increasingly dense and accurate reconstruction through the multiple stages. Note that, the ground truth point cloud is obtained by scanning, which is even of lower quality than our reconstructions in this example.

and learns to predict similar uncertainty. Our uncertainty-aware volume construction process enables highly efficient spatial partitioning, which further allows for the final reconstruction to be of high accuracy and high completeness.

Evaluation of multi-stage depth prediction. We have quantitatively demonstrated that our multi-stage framework reconstructs scene geometry with increasing accuracy and completeness in every stage (see Fig. 2.3). We now further evaluate our network and do ablation studies about different stages on the DTU testing set with detailed quantitative and qualitative comparisons. We compare with naive upsampling to justify the effectiveness of our uncertainty-aware coarse-to-fine framework. In particular, we compare the results from our full model and the results from the first two stages with naive bilinear upsampling using a scale of 2 (for both

Table 2.7. Performance comparisons. We show the running time and memory of our method by running the first stage, the first two stages and our full model.

Method	Running time (s)	Memory (MB)	Input size	Prediction size
One stage	0.065	1309	640x480	160x120
Two stages	0.114	1607		320x240
Our full model	0.257	1647		640x480
MVSNet [177]	1.049	4511	640x480	160x120
R-MVSNet [178]	1.421	4261	640x480	160x120

height and width) in Tab. 2.6. We can see that upsampling does improve the reconstruction, which benefits from denser geometry and using our high-quality low-resolution results as input. However, the improvement made by naive upsampling is very limited, which is much lower than our improvement from our ATV-based upsampling. Our UCS-Net makes use of the ATV – a learned local spatial representation that is constructed in an uncertainty-aware way – to reasonably densify the map with a significant increase of both completeness and accuracy at the same time.

Figure. 2.5 shows qualitative comparisons between our reconstructed point clouds and the ground truth point cloud. Our UCS-Net is able to effectively refine and densify the reconstruction through multiple stages. Note that, our MVS-based reconstruction is even more complete than the ground truth point cloud that is obtained by scanning, which shows the high quality of our reconstruction.

Comparing runtime performance. We now evaluate the timing and memory usage of our method. We run our model on the DTU validate set with an input image resolution of $W \times H = 640 \times 480$; We compare performance with MVSNet and R-MVSNet with 256 depth planes using the same inputs. Table 2.7 shows the performance comparisons including running time and memory. Note that, our full model is the only one that reconstructs the depth at the original image resolution that is much higher than the comparison methods. However, this hasn't introduced any higher computation or memory consumption. In fact, our method requires significantly less memory and shorter running time, which are only about a quarter of the memory and time used in other methods. This demonstrates the benefits of our coarse-to-fine framework with fewer depth planes (104 in total), in terms of system resource usage. Our UCS-Net with ATVs achieves high-quality reconstruction with high computation and memory efficiency.

2.5 Summary

In this chapter, we present a novel deep learning-based approach for multi-view stereo. We propose the novel uncertainty-aware cascaded stereo network (UCS-Net), which utilizes the adaptive thin volume (ATV), a novel spatial representation. For the first time, we make use of the uncertainty of the prediction in a learning-based MVS system. Specifically, we leverage variance-based uncertainty intervals at one cascade stage to construct an ATV for its following stage. The ATVs are able to progressively sub-partition the local space at a finer scale, and ensure that the smaller volume still surrounds the actual surface with a high probability. Our novel UCS-Net achieves highly accurate and highly complete scene reconstruction in a coarse-to-fine fashion. We compare our method with various state-of-the-art benchmarks; we demonstrate that our method is able to achieve the qualitatively and quantitatively best performance with high computation- and memory- efficiency.

We have presented a deep MVS approach to reconstruct the geometry in a real scene; our novel UCS-Net takes a step towards making the learning-based geometry reconstruction more reliable and efficient. In the following chapters, we explore other appearance acquisition problems, including material capture and image-based appearance acquisition.

This chapter is a reformatted version of the material as it appears in “Deep Stereo using Adaptive Thin Volume Representation with Uncertainty Awareness,” Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, Hao Su. [25]. The material has been submitted and accepted to the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. The dissertation author was the co-primary investigator and author of this paper.

Chapter 3

Reflectance Acquisition from Sparse Optimal Samples

3.1 Introduction

Geometry models the global structures of a scene, which describes where the objects are in the scene; we have introduced a novel learning based MVS method that can efficiently reconstruct high-quality scene geometry in Chapter 2. On the other hand, materials model the local appearance of a scene, which depict how any point at any object in the scene looks like. It is the geometry and materials that complete the entire scene appearance under given illumination and viewpoint. In this chapter, we study the problem of capturing complex real materials. In particular, we acquire the reflectance of real materials which models how light scatters at the surface of the material and determines the material appearance.

The reflectance of an opaque material is modeled by a bidirectional reflectance distribution function (BRDF). The BRDF is a 4D function that defines how light is reflected at the surface of an opaque object, given any incident (2D) and exitant (2D) directions. Accurate BRDF models are critical for realistic image synthesis. Many analytic BRDF models have been proposed [147, 157]. However, the greatest fidelity is obtained by data-driven reflectance, such as the MERL BRDF database of 100 real materials [100].

In this chapter, we focus on the canonical problem of measuring the 3D isotropic BRDF of a flat sample of homogeneous material. The conventional approach is to use a gonioreflectometer,

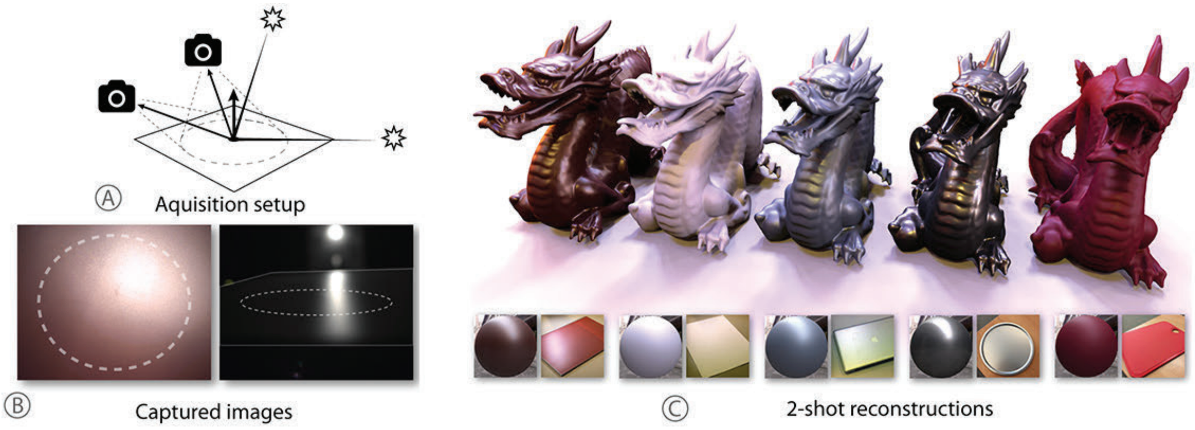


Figure 3.1. By utilizing the field of view of a camera (A), many radiance observations can be acquired in parallel, to enable efficient BRDF measurement from a homogeneous flat material sample (we assume a near-field camera and distant light source). We demonstrate that as little as 2 view/light configurations enable accurate reconstruction for most materials. The two input images are shown in (B); our acquisition uses data from the circular center of the sample outlined, which may appear elliptical at oblique angles. Qualitatively, the first image captures the overall shape and intensity of the specular highlight and diffuse color. The second image captures grazing angles and Fresnel effects, also refining diffuse shading. From these inputs, we reconstruct a full measured BRDF, which can be visualized on a sphere or used for rendering (C); input material samples are shown in the insets and in results in Fig. 3.11.

laboriously sampling illumination-view pairs [41]. However, fully sampling a 3D isotropic BRDF domain can require thousands or millions of samples, making this approach very expensive. Mirror-based imaging setups [157] can reduce some dimensions, but still require a large number of samples. They also need more complex setups, and can be difficult to calibrate.

Recently, Nielsen et al. [109] presented a significant reduction in the number of samples needed to about 20, assuming the BRDF lies approximately in the subspace of the MERL BRDF database. They leverage a logarithmic mapping (originally proposed for BRDF factorization by [102]). They then optimize sampling directions to minimze the condition number of the related acquisition matrix (an approach first proposed by [101]).

However, there are several limitations of [109]. First, they use a gantry-based system, where each "measurement" is actually a full 2D image seen by the camera. However, this additional information is not used in their work, providing only a single observation. We seek to exploit the additional degrees of freedom by acquiring multiple BRDF measurements simultaneously, using a near-field camera, so each point on the sample corresponds to a slightly different viewing direction. In this chapter, we demonstrate an improved minimal BRDF sampling method for near-field acquisition. Indeed, accuracy comparable to the 20 measurements in [109] is achievable with only two near-field images, and high-fidelity results are sometimes achieved with a single image, with field-of-view only about 25°, as shown in Figs. 3.1, 3.2.

A major technical challenge is finding the optimal light-view directions. The conventional condition number metric is not adequate, since it can increase dramatically (or even go to infinity) for a set of closely-related near-field measurements. While the measurements are no longer completely independent, they do provide additional information. Even for the goniometric case in [109], we show that condition number does not fully model the error.

We therefore develop an entirely new framework to accurately estimate the error in BRDF acquisition from a set of samples, considering both deviation from the ideal (noise, BRDF not fitting MERL data), and accuracy of reconstruction based on where samples are located (Fig. 3.3). The condition number only approximates error from the first term (noise), while we usually want to minimize the latter term (reconstruction error). Our framework enables significantly better reconstruction error for the multiple-measurement near-field setup (Fig. 3.5). We make the following contributions:

Novel Theoretical Analysis: We develop a new framework to predict the error in BRDF acquisition from sampling a particular set of directions (Sec. 3.4), that extends naturally to multiple measurements and allows for optimization (Sec. 3.5).

Optimal BRDF Sampling Directions: We provide optimal light-view directions for the near-field case (Sec. 3.6), showing how multiple simultaneous BRDF measurements can dramatically reduce the number of images (see Fig. 3.2 for simulations on MERL database). The new error framework also provides improved accuracy even for point-sampled measurement (Appendix C).

Practical BRDF Acquisition: We develop BRDF measurement using only two near-field images (two-shot), and we demonstrate results on several real examples (Sec. 3.7). We also

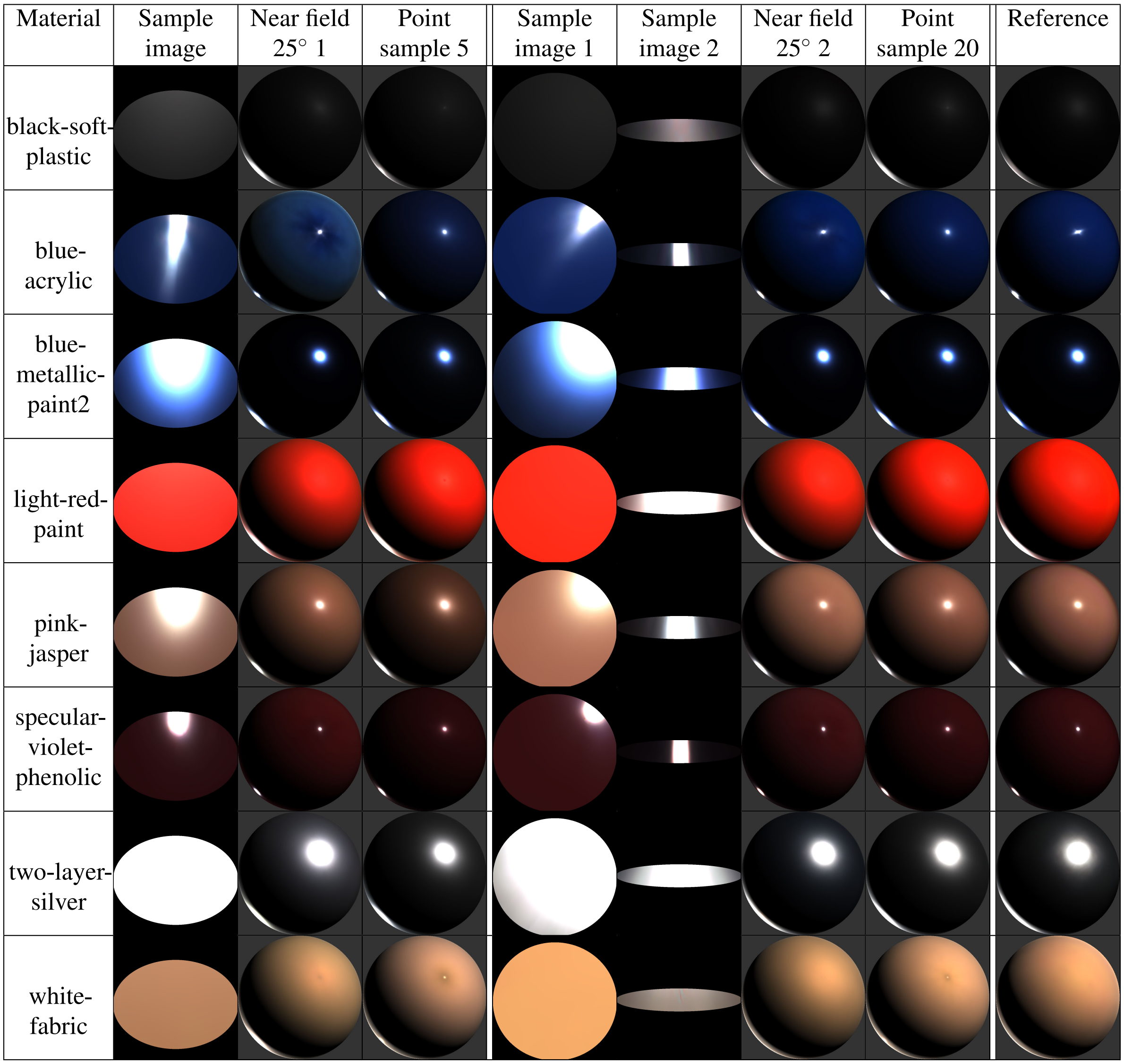


Figure 3.2. Simulations on MERL BRDF database for one and two-shot near-field BRDF measurement. BRDFs shown are not used at all in analysis/choosing the optimal directions. The sample subtends an angle of 25° when camera is at the zenith (we consider a circular region, sample images elliptical when viewed at an oblique angle; see Figs. 3.4, 3.9). After sample label, first 3 columns show single image near-field acquisition, with input, our method and previous work with 5 point samples. The next 4 columns show two image acquisition with inputs, our method, and previous work with 20 point samples. The reference image is shown rightmost. Single-shot measurement is comparable to 5 samples in [109]. Near-reference quality images, comparable to 20 samples in [109], are obtained with two-shot meaurement. For very broad specularities like light red paint, we do not fully observe the highlights in the input images, and therefore slightly underestimate their width. The supplementary material has results for simulations on all of the MERL BRDF materials.

briefly explore extensions to a fixed camera setup with two light directions (Sec. 3.8), including a simple first step towards acquiring spatially-varying BRDFs by clustering materials.

3.2 Related Work

We focus on measured data-driven BRDFs. A few recent works have evaluated and developed improved parametric fits to the MERL BRDF database [108, 95, 12]. Earlier, Lensch et al. [86] proposed adaptive sampling of a BRDF, based on the uncertainty of the parametric fit. Fuchs et al. [43] also proposed an adaptive sampling approach, but first require a relatively dense grid of samples. Our approach actually requires less data (only one or two images), is more accurate, and requires only linear solution, rather than non-linear optimization. Our method is also conceptually related to other areas, such as the use of key points for animation [104]. We briefly review related work in efficient BRDF acquisition below.

Basis and Environment Lighting: One approach to speed up BRDF acquisition is to use basis functions for complex illumination [52, 3, 149]. Our setup is simpler and requires fewer measurements, enabling direct BRDF measurement from a directional source, without any deconvolution. Other work has dealt with (uncontrolled) environment lighting [126, 127], but requires non-linear regularization and priors.

Image-Based BRDF Acquisition: The MERL database was acquired by Matusik et al. using image-based BRDF measurement [98] on spheres. Our approach is conceptually similar, making use of images rather than point measurements, but works with conventional flat surfaces (not all BRDFs are easily found as, or can be wrapped on, spheres). Earlier, Karner et al. [77] fit anisotropic BRDFs to images of a flat sample, but required a parametric model. A variety of other optical setups acquire multiple samples simultaneously, including [157, 28, 111]. These methods usually require complex imaging setups, which are hard to calibrate.

One and Two Shot Approaches: Most recently, Aittala et al. [4] proposed a two-shot spatially-varying BRDF capture setup, but it is aimed at reproducing the “texture” of material samples, rather than a complex measured BRDF. Earlier, [60] used reference BRDFs to recover shape and spatially-varying reflectance. Ren et al. [123] developed a pocket reflectometry method, comparing to reference tiles, using a handheld light source and fixed camera. In contrast, we do not require a physical reference since we can leverage the MERL database, which also has a much broader set of reference materials.

Industry Material Standards: Beyond computer graphics, the materials industry has developed a number of standards for measuring and characterizing reflectance. [66] proposes a single measurement at 60° perfect reflection. The ASTM Standard D523 for measuring gloss adds near-normal and grazing angle measurements at 85° and 20° [67]. We extend this approach by considering near-field measurements. Our optimal sets of one and two measurements produce results that are close to the above observations, but are based on rigorously minimizing the expected error. Moreover, we can recover a full accurate data-driven BRDF, since we consider multiple measurements over the entire sample. Additional works in graphics include the five measurement directions in [160], which are improved on by [109] and further refined by our method.

3.3 Background

In this section, we briefly discuss necessary background on using the MERL [100] database for BRDF measurement, following [109]. We conclude by providing intuition for why the condition number metric is not ideal, especially in the near-field lighting case, a result that may also impact other problems involving sparse sampling and reconstruction in graphics.

BRDF Database and Processing: The database consists of 100 materials. Each material is represented using $p = 1,458,000$ exhaustive measurements of the 3D isotropic BRDF volume in the $(\theta_h, \theta_d, \phi_d)$ parameterization [130] (Fig. 3.4), with resolution $90 \times 90 \times 180$ degrees. Following [109], we treat each color channel separately, effectively obtaining $m = 300$ database BRDFs. We also apply their log-relative mapping (inspired by the logarithmic transform proposed in [102]). BRDF ρ is transformed to $\ln[(\rho w+\varepsilon)/(\rho_{\text{ref}}w+\varepsilon)]$, where $\varepsilon = 0.001$ avoids division by zero, ρ_{ref} is the reference or (per-observation) median BRDF, and the weight w is simply the maximum of the cosine of incident and outgoing angles. In this chapter, we deal only with these log-mapped BRDFs. Inverse mapping is done at the end to obtain the final measured BRDF.

BRDF Principal Components: Let $X \in \mathbb{R}^{m\times p}$ be the full matrix of all MERL BRDF observations, where the rows are mapped BRDFs and the columns are a particular direction. We use the principal components Q, obtained by performing a singular-valued decomposition (SVD) after subtracting the mean BRDF,

$$X - \hat{\mu} = U\Sigma V^T \qquad Q = V\Sigma, \tag{3.1}$$

where it is convenient to include the singular values in $Q \in \mathbb{R}^{p\times k}$. k is the number of principal components we consider (in our case, $k = m$, but one could use fewer components). The columns of Q are the scaled eigenvectors of the covariance, and correspond to a basis for the space of BRDFs. A particular BRDF x may be obtained as a linear combination of the basis,

$$x = Qc + \mu, \tag{3.2}$$

where $c \in \mathbb{R}^{k\times 1}$ is a vector of coefficients. $\mu \in \mathbb{R}^{p\times 1}$ is the mean BRDF, while $\hat{\mu} \in \mathbb{R}^{m\times p}$ is a matrix, repeating μ^T over m rows.

Solving for the Measured BRDF: In practice, we observe x at some sample observations, from which we seek to estimate c,

$$\tilde{x} - \tilde{\mu} = \tilde{Q}c, \tag{3.3}$$

where the tildes indicate that we have a reduced set of observations at n samples, with $\tilde{\mu}$ and

$\tilde{x} \in \mathbb{R}^{n\times 1}$. $\tilde{Q} \in \mathbb{R}^{n\times k}$ is the set of rows in Q corresponding to the set of reduced observations. It is also convenient to define $y = x - \mu$, with $\tilde{y} = \tilde{x} - \tilde{\mu}$ and $\tilde{Q}c = \tilde{y}$. Finally, let $S \in \mathbb{R}^{n\times p}$ be a selection matrix that is zero everywhere, except that $S_{ij} = 1$ in row i iff j is the direction corresponding to observation i. We can now define the reduced $\tilde{Q} = SQ$ and $\tilde{y} = Sy$, which will be useful for the error analysis in Sec. 3.4.

In [109], $n \ll p,m$, and typically $n \sim 20$. In our case, for near-field imaging, we have fewer image captures (typically only one or two), but we have several observations at each image, since we make use of the full 2D image. n can now be larger and, in some cases, could even be greater than m. However, the near-field samples are correlated, having the same light and similar view directions, so conceptually we still have a reduced matrix.

The above equation can be solved for the coefficients using Tikhonov regularization (I is the identity matrix. We set $\eta = 40$; we find results are not sensitive to this regularization parameter),

$$c = \text{argmin}|\,(\tilde{x}-\tilde{\mu}) - \tilde{Q}c|^2 + \eta|c|^2 = \left(\tilde{Q}^T\tilde{Q} + \eta I\right)^{-1}\tilde{Q}^T\tilde{y}. \tag{3.4}$$

A useful intuition is to consider a closed-form expression for the regularized inverse. Assuming the full SVD of $\tilde{Q} = A\Lambda B^T$,

$$\tilde{Q}^+_\eta = \left(\tilde{Q}^T\tilde{Q} + \eta I\right)^{-1}\tilde{Q}^T = B\Lambda^+_\eta A^T, \tag{3.5}$$

where Λ^+_η is a diagonal matrix with the same shape as Λ, but with a modified set of singular values: $\sigma \to \sigma/(\sigma^2+\eta)$. Note that the pseudo-inverse $\tilde{Q}^+$ and Λ^+ are obtained by setting $\eta = 0$, in which case $\tilde{Q}^+ = B\Lambda^+A^T$ as expected. In essence, the regularization term creates a $\eta-$modified pseudo-inverse where the inversion of small singular values does not blow up.

Optimizing Sampling Directions: In [101, 109], the optimal sampling directions (the rows of $\tilde{Q}$ chosen, or equivalently the selection matrix S with $\tilde{Q} = SQ$) are found by optimizing (minimizing) the condition number,

$$\kappa(\tilde{Q}) = \frac{\sigma_{\mathrm{m}ax}(\tilde{Q})}{\sigma_{\mathrm{m}in}(\tilde{Q})}, \tag{3.6}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the maximum and minimum singular values of $\tilde{Q}$. The condition number is a standard numerical tool, and reducing it minimizes the sensitivity to noise and related errors.

Formally, consider a matrix equation such as equation 3.3, with $\tilde{Q}c = \tilde{y}$ (with $\tilde{y} = \tilde{x} - \tilde{\mu}$ as usual). The condition number is the worst case (upper bound) estimate of the ratio of fractional error δc in output to fractional error/noise $\delta\tilde{y}$ in input,

$$\frac{|\delta c|/|c|}{|\delta\tilde{y}|/|\tilde{y}|} \leq \kappa(\tilde{Q}). \tag{3.7}$$

3.3.1 Limitations of Using the Condition Number

The condition number gives good results for point-sampled BRDF measurement [101, 109]. It can be considered a measure of correlation between samples, and minimizing it chooses sampling directions that discriminate between distinct BRDFs. However, we found that it did not easily extend to near-field measurements, where a large number of related observations are made (Fig. 3.5). The observation matrix $\tilde{Q}$ is now often rank deficient or nearly so, and close-by observations can drive the condition number very large or even to infinity, reducing its ability to discriminate and choose optimal directions. This leads to the paradox where fewer observations are preferred. In the next section, we formally derive the expected error, considering both reconstruction error and noise. For near-field BRDF measurement, we achieve a dramatic improvement; one to two near-field images is adequate.

There are also many technical limitations of condition number. First, there are two terms related to error: noise or other imperfections (deviations from MERL data); and reconstruction error caused by having too few samples (even in the presence of zero noise or deviation).

$\kappa(\tilde{Q})$ only bounds the first term (noise/deviations), but the major component of the error is actually reconstruction error from having fewer observations than principal components. Second, condition number considers fractional error, assuming the error is proportional to the signal. However, the accuracy of measurements from real cameras is determined by a number of factors (shot noise, read noise, dark current), which are constant or proportional to the square root of intensity, and not the intensity itself. Indeed, well lit pixels have less relative noise, and in this chapter we more accurately model the noise as a constant magnitude, independent of the signal. Third, κ only provides a worst-case bound, while we are often interested in the average error, say over all of the materials in the MERL BRDF database. Hence, our optimal sampling directions improve somewhat on [109] even for point-sampling.

3.4 Sampling Error Analysis

In this section, we conduct a novel analysis of the BRDF reconstruction error from a sparse set of samples. This error can be minimized to find the optimal set of sampling directions, for both conventional point-wise BRDF acquisition, and near-field image-based measurement. For completeness, we consider three sources of error: deviation from the MERL database, sparse sampling, and noise in measurement. In practice, deviation error from the MERL database is not easy to predict, nor is the real noise level easy to evaluate. Therefore, our practical algorithm will focus on minimizing the reconstruction error from sparse sampling, which is the main factor in choosing suitable directions for BRDF acquisition.

Deviation from BRDF Model: We assume the BRDF being measured lies in the subspace spanned by the MERL database (and encapsulated in Q). If this is not the case, we can only find the best projection of the MERL BRDF data. This error is present even when we have all observations. Using pseudo-inverse Q^+ of Q,

$$c = Q^+(x-\mu) = Q^+y = \Sigma^{-1}V^Ty, \tag{3.8}$$

where we expand $Q = V\Sigma$. The resulting deviation error is,

$$\boxed{E_{\text{deviation}} = |Qc - y| = |(VV^T - I)y|.} \tag{3.9}$$

Note that $V \in \mathbb{R}^{p\times k}$ is an orthogonal matrix with $V^TV = I$, but since $k < p$, $VV^T \in \mathbb{R}^{p\times p}$ is not the identity. However, if y is in the MERL BRDF database, it is given as a column of $Y^T = (X - \hat{\mu})^T = V\Sigma U^T$. Using the SVD decomposition, it is easy to see that $(VV^T - I)V\Sigma U^T = 0$, since $V^TV = I$.

Therefore, $E_{\text{deviation}} = 0$ if the material is in the subspace Q spanned by the MERL database, but will be nonzero if it lies outside this subspace. This is an intrinsic property of the material, and *independent of the sampling directions chosen.*

Projection to Sampling Directions: Choosing a sparse set of n sampling directions corresponds mathematically to choosing a particular selection matrix $S \in \mathbb{R}^{n\times p}$. Noting that $\tilde{Q} = SQ$ and $\tilde{y} = Sy$ by definition, so that $SQc = Sy$, we have

$$\bar{c} = (SQ)^+_\eta (Sy), \tag{3.10}$$

where in the last line we consider the regularized inverse of SQ, as per equation 3.5, and Sy are the observations we actually make with a camera or a gonioreflectometer. We use the bar on top of c to denote the recovered coefficients, with error

$$c - \bar{c} = \left(Q^+ - (SQ)^+_\eta S\right) y. \tag{3.11}$$

Finally, the reconstruction error is given by

$$\boxed{E_{\text{recon}} = \left| Q\left(Q^+ - (SQ)^+_\eta S\right) y \right|.} \tag{3.12}$$

This is the critical error we need to minimize, by choosing sampling directions (and hence S)

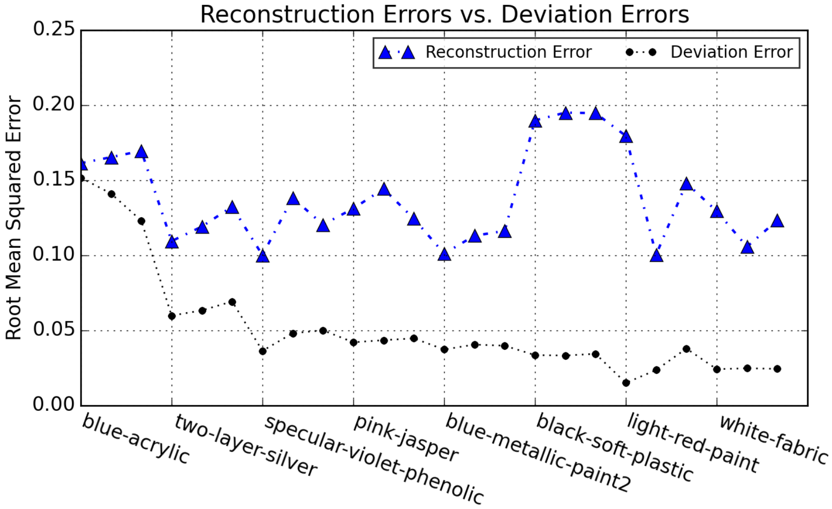


Figure 3.3. Comparison of E_{recon} and $E_{\text{deviation}}$ for all materials in Fig. 3.2. Reconstruction error E_{recon} is dominant for most BRDFs.

optimally. It provides the error in reconstruction by measuring only a sparse set of samples, and applies equally whether those are point samples or multiple simultaneous image-based measurements. Note that this error exists even for noise-free measurements, coming purely from reconstruction error when using a sparse set of samples. (By using log-mapped BRDFs, we also limit the ability of intense specularities to unduly influence reconstruction error.) The condition number does not consider this term directly, but only sensitivity to noise. Nevertheless, we show in Appendix B that minimizing condition number does adjust SQ to reduce (but not minimize) E_{recon}.

Figure 3.3 shows both deviation and reconstruction errors for the BRDFs in Fig. 3.2 (using a different set of 90 MERL materials as our data/training set). As expected, reconstruction error E_{recon} dominates in all cases. Blue acrylic has high $E_{\text{deviation}}$ since the star-shaped highlight deviates significantly from the database.

Noise Error: If we do have noisy data, the image observations y will be corrupted, and we will measure $\bar{y} = y + \triangle$, where $\triangle$ is the noise or error at each pixel. The resulting error in the coefficients is given from equation 3.10 by $(SQ)^+_\eta(S\triangle)$. Therefore,

$$\boxed{E_{\text{noise}} = \left|Q\,(SQ)^+_\eta\,S\triangle\right|.} \tag{3.13}$$

Conceptually, the condition number seeks to minimize this term. However, condition number provides only a worst-case bound, assuming the noise is proportional to the signal, which is not a correct assumption for cameras, where noise levels are relatively independent of image intensity. Moreover, our main focus is on reconstruction error from sampling (equation 3.12) rather than noise; one typically acquires high-dynamic range images from high-end cameras where noise is not the most significant challenge. Note that the condition number analysis also does not consider the full process, including the η-regularization. Finally, our focus is on near-field capture where we have several, but closely-related observations. This can lead to a very large condition number, while in fact the additional observations help in reducing the error.

The total error is written simply as (the less than sign comes from the triangle equality, since each error term considers the norm),

$$E_{\text{total}} \leq E_{\text{deviation}} + E_{\text{recon}} + E_{\text{noise}}. \tag{3.14}$$

Final Error Metric: For simplicity, we do not explicitly consider the deviation error, but just include it as part of the noise/error $\triangle$. A final issue is choosing y in equation 3.12 and $\triangle$ in equation 3.13, since these quantities depend on the measurements, camera noise, and are not known a-priori. For y, we minimize over all of the m materials in the MERL BRDF, essentially finding the sampling directions that best reconstruct the MERL materials. Define $y_i = x_i - \mu$ where $x_i \in \mathbb{R}^{p\times 1}$ is a vector corresponding to observations of BRDF i in the MERL database. For the noise, we assume a constant user-defined parameter β, corresponding to the noise/error level, $\beta = |\triangle|$, and use a noise vector $O \in \mathbb{R}^{p\times 1}$, where each element is simply 1. This can be seen as the expected magnitude of gaussian-distributed noise, where β controls the magnitude.

Putting this together, our final expected error is,

$$\boxed{E(S) = \left(\frac{1}{m} \sum_{i=1}^{m} \left| Q \left(Q^{+} - (SQ)_{\eta}^{+} S \right) y_i \right| \right) + \beta \left| Q\,(SQ)_{\eta}^{+}\, SO \right| .} \tag{3.15}$$

Note that we add errors from reconstruction and noise. Each term on the right-hand side is a $p \times 1$ vector, and we take its norm. We also make explicit the dependence of E on selection matrix S.

In this chapter, we focus mainly on minimizing the reconstruction error E_{recon} by choosing sampling directions. Therefore, we typically take $\beta = 0$, but we also demonstrate nonzero noise β in supplementary material. Finally, we emphasize that we have so far only defined error; the next section discusses how to choose the sampling directions, corresponding to the selection matrix S, to minimize this expected error. In essence, we seek $S = \operatorname{argmin} E(S)$.

3.5 Optimal Sampling Directions

We first describe selection of the optimal sampling directions for measuring individual light-view pairs in a BRDF, as in [109]. We refer to this as point sampling, to distinguish from the near-field image-based BRDF measurement of our method.

3.5.1 Point-Sampled BRDF Directions

We consider the whole space of valid directions $\mathscr{D} = \{\theta_h, \theta_d, \phi_d\}$ in the MERL database. Our goal is to find the optimal subset $\mathscr{D}^n$ with n directions, and form the corresponding n-row selection matrix S^n. For point samples, each row in S^n is simply one direction in $\mathscr{D}^n$. In other words, $S^n_{ij} = 1$ iff $\mathscr{D}^n_i = j$. The optimal $\mathscr{D}_n$ (and S_n) must be chosen to achieve the minimal error in equation 3.15.

We solve the optimization based on a numerical gradient descent framework analogous to [109], which is shown in that work to be more efficient and higher-quality than the greedy method of [101]. (Standard numerical optimizers do not work well, given the discrete BRDF

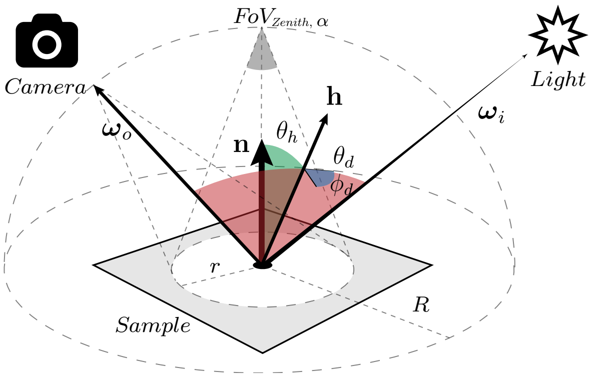


Figure 3.4. Schematic of near-field reflectance acquisition. The half-diff angles $\theta_h, \theta_d, \phi_d$ are with respect to the center of the sample.

space $\mathscr{D}$ and integer steps needed, as well as invalid BRDF regions.) However, we replace the condition number with the accurate error in equation 3.15, and extend the optimization framework to near-field measurements in Sec. 3.5.2. We start with an empty set $\mathscr{D}^0$ with no directions in it. Then we iteratively extend $\mathscr{D}^n$ to $\mathscr{D}^{n+1}$ as follows:

1. Randomly pick t candidate directions from $\mathscr{D} - \mathscr{D}^n$ (Typically we use t = 500.). For each candidate direction d, we form a selection matrix S_d of $\mathscr{D}^n \cup d$, and evaluate the expected error $E(S_d)$ from equation 3.15. An initial $\mathscr{D}^{n+1}$ and S^{n+1} is created with that $\mathscr{D}^n \cup d$ which has a minimal $E(S_d)$.

2. Randomly choose one of the $n+1$ directions in $\mathscr{D}^{n+1}$, which we denote as $(\theta_h, \theta_d, \phi_d)$. Estimate the gradient of the error metric $\bigtriangledown E(S^{n+1}) = (\frac{\delta E(S^{n+1})}{\delta \theta_h}, \frac{\delta E(S^{n+1})}{\delta \theta_d}, \frac{\delta E(S^{n+1})}{\delta \phi_d})$ numerically. Move the chosen direction along $\bigtriangledown E$ with one step-length (initial step-length is 3 cells) if the destination is a valid location in $\mathscr{D}$. Repeat until convergence (finding a new direction each time).

3. Reduce step-length and repeat step 2 until convergence with step of 1 cell. Then the final $\mathscr{D}^{n+1}$ and S^{n+1} are formed.

3.5.2 Near-Field BRDF Directions

We now take advantage of sampling directions for all pixels in an image, instead of only the center of the image for point sampling. In general, the optimal directions depend on the camera's projection matrix and the size of the planar sample. To develop a general framework, we assume the image of interest is a circle on the plane with radius r. We assume the camera moves on a hemisphere a distance R from the center of the circle, and is always pointed towards the center (i.e., the center pixel corresponds to the center of the sample). We also assume the image always sees the full sample (circle of interest). The key variable is the ratio $\nu = r/R$, which determines the linear field of view when the camera is at the zenith. The angular field of view $\alpha = 2\tan^{-1}\nu$, which is the angle we use to denote our near-field setup. A schematic of the setup is shown in Figs. 3.1 and 3.4. Note that the optimization framework is general, and can also apply to many other configurations. We discuss one-camera multiple light and one-light multiple view cases in Sec. 3.8.

The goal is still to find an optimal subset of camera directions $\mathscr{D}^n$. In this case, each direction represents the direction to the camera with respect to the center pixel. However, the corresponding selection matrix is no longer a n-row matrix. We replace S^n with $\bar{S}^n$ in near field acquisition. One direction in $\mathscr{D}^n$ forms a set of rows in $\bar{S}^n$, each of which corresponds to one pixel sample in an image. In general, there will be many more rows than for point-sampling, but many of the directions will be very closely related. Our error metric addresses this directly, and equation 3.15 still accurately predicts reconstruction error. We can now directly use $\bar{S}^n$ instead of S^n, and iteratively add directions from $\mathscr{D}^0$ to $\mathscr{D}^n$ as before. To validate the convergence of our method, we repeated the optimization 50 times with different random conditions, and fields of view. The results all converge well. The supplementary material shows convergence results for $n = 2$ near-field sampling with 25° field of view.

3.6 Evaluation with Simulations for Near-Field

We now evaluate the minimization of our error metric for near-field image-based BRDF measurement, using simulations with the MERL BRDF database. As shown in Fig. 3.5, our new error metric has significant advantages over using the condition number for the near-field case. (Visual results on rendered spheres are consistent with these numerical errors; results from minimizing condition number are often even worse than point-sampled measurements, far off from ground truth). Moreover, as seen in Fig. 3.6, our new image-based method is much more efficient than point-sampling; both methods capture similar images of a flat sample, but we make use of the full 2D image. Section 3.8 applies the framework to other configurations like fixed camera with multiple lights, or vice versa.

The setup is shown in Fig. 3.4. For simplicity, we assume a distant light source, and a near-field camera. In a single image, we capture a 2D slice of the BRDF (we consider a circular patch). Since we are assuming a flat sample with distant lighting, the illumination direction is the same everywhere, but the viewing direction varies at each pixel, enabling us to capture multiple observations simultaneously. It is clearly better to have a wider field of view to capture greater view variation, but this may require large samples and close cameras. In fact, we show that a relatively narrow field of view of about 25° suffices for two-shot BRDF acquisition.

We minimize equation 3.15, choosing the optimal light-view directions, as explained in Sec. 3.5.2. To evaluate the reconstruction error on the MERL BRDF, we use a slightly different set of directions using 90 training BRDF samples, testing on the 10 other materials not used at all in computing optimal directions. (We use the same training/test materials as [109] to enable direct comparisons to their approach.)

Figure 3.5 compares our average normalized reconstruction RMS errors for the unknown materials for several fields of view, as a function of the number of images, and to optimizing condition number alone. As shown in Appendix C, condition number is actually a reasonable heuristic for point-sampled BRDF measurement [109], although our error metric performs

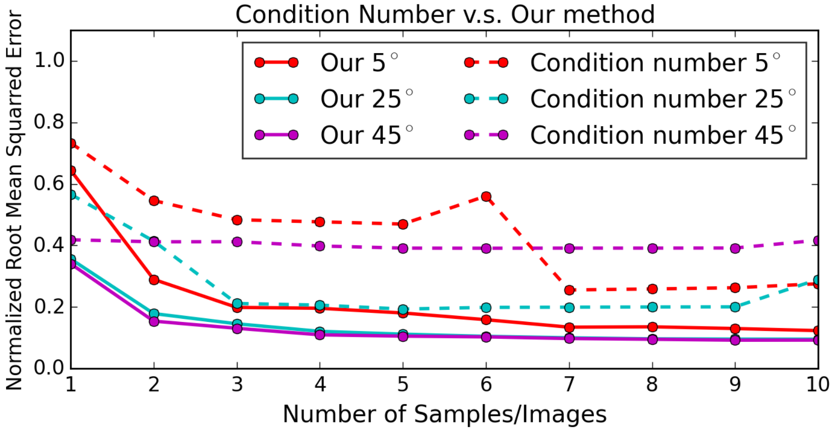


Figure 3.5. Comparison of errors on unknown samples from our method, and from minimizing condition number, for near-field reflectance acquisition with different fields of view. It is clear that we produce better results for near-field reflectance acquisition.

somewhat better even in that case. However, it breaks down for near-field acquisition as seen in Fig. 3.5. With several correlated view directions, condition number becomes very large, losing the ability to discriminate between different sets of light-view directions. In some cases, it oscillates or does not decrease with increasing samples, while our method always performs well. *The new error metric is essential for determining optimal light-view directions in the near-field case.*

In Fig. 3.6, we compare RMS errors for several fields of view, and to point-sampled BRDF measurement (the top red curve is from [109] while the improved orange curve is using our error metric for the point-sampled case). We see that near-field reflectance acquisition requires almost an order of magnitude fewer images than point-sampled BRDF measurement. Also note that near-field acquisition converges quickly with increasing field of view; while larger fields of view help, 25° is already nearly best (supplementary shows similar curves even for extreme 85° and 175° fields of view). In fact, errors are comparable to standard spherical image-based BRDF measurement [98] (with optimal directions chosen by our error metric; see Fig. C.2 in Appendix C). However, our approach applies more generally, to flat samples that cannot be obtained or wrapped on a sphere.

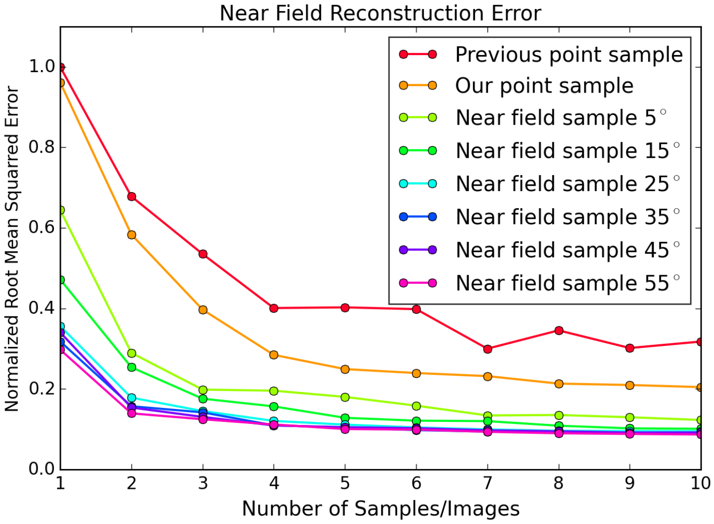


Figure 3.6. Average RMS error over unknown samples for near-field reflectance acquisition. We plot the results for a number of different field of view angles. These results clearly show the benefits of our method, often requiring an order of magnitude fewer samples than point-sampled BRDF measurement.

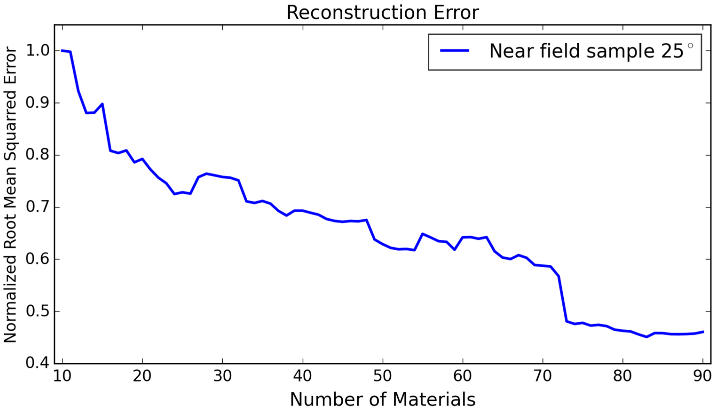


Figure 3.7. Average RMS error versus number of materials in database for 2 shot near-field sampling with 25° field of view.

Figure 3.7 shows how errors decrease as more training materials are added to the database (in random order), showing a steady decrease with more example BRDFs. Note that the MERL subspace depends on the specific BRDFs used, and the error curve can therefore increase slightly with the addition of a new material.

Note that with two-shot imaging, we can obtain essentially the same accuracy as 20 samples for point-sampled BRDF measurement, and even single-shot near-field acquisition

n	$\theta_h[^\circ]$	$\theta_d[^\circ]$	$\phi_d[^\circ]$
1	3	42	1
2	0	60	0
	34	36	32

(a) 15°

n	$\theta_h[^\circ]$	$\theta_d[^\circ]$	$\phi_d[^\circ]$
1	3	50	9
2	6	23	41
	16	79	87

(b) 25°

n	$\theta_h[^\circ]$	$\theta_d[^\circ]$	$\phi_d[^\circ]$
1	3	52	0
2	4	40	0
	8	81	82

(c) 35°

n	$\theta_h[^\circ]$	$\theta_d[^\circ]$	$\phi_d[^\circ]$
1	3	60	4
2	0	81	0
	5	40	4

(d) 45°

Figure 3.8. Tabulation of one and two near-field acquisition directions for fields of view ranging from 15° to 45°. Note that directions correspond closely to mirror reflection, imaging the highlight shape, and more grazing angles for Fresnel and other effects.

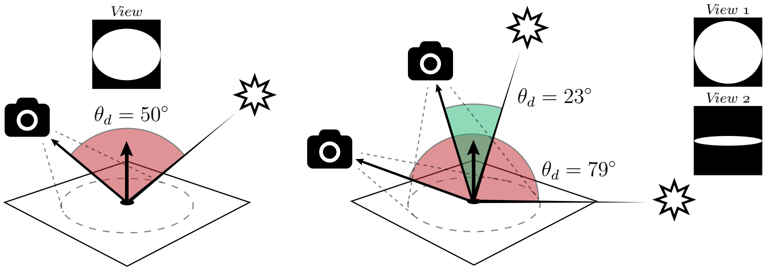


Figure 3.9. Illustration of one- and two-shot camera/light configurations for a linear field of view of 25°.

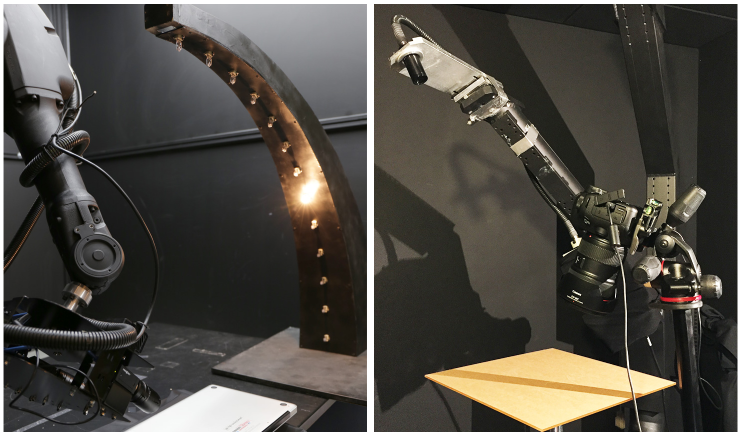
achieves similar accuracy as about 5 samples for point-sampled BRDF measurement. These comparisons, simulated on the MERL data, are shown in Fig. 3.2. (We omit green fabric and silver metallic paint, whose results are very similar to black soft plastic and two layer silver respectively). We see that two-shot acquisition is adequate in nearly all cases. In some examples like pink jasper, specular violet-phenolic and white fabric, a single near-field image is comparable to 20 point-sampled images. In a few cases, like blue acrylic, two near-field images are required to achieve reasonable results; severe ringing is present in reconstruction from a single image. For very broad specularities like light red paint, we do not observe the full extent of the highlight in any single image, and slightly underestimate highlight width.

Finally, in Fig. 3.8, we tabulate our optimal 1,2 directions for near-field angles of 15°, 25°, 35° and 45°. Note that these directions are with respect to the center of the sample; the local view direction will vary at each pixel. For one image, we capture a slightly off-specular direction ($\theta_h = 3°$) at an angle of incidence of about 50°. Similarly, for two images, the first direction for 15° field of view is an exact specular reflection at 60°, although this varies somewhat with field of view. This is as expected, imaging the details of the specular highlight, around the center of the sample, and also accords well with measurements previously used in the appearance industry [66]. For most materials, this measurement also captures the overall diffuse color. The second direction usually varies somewhat from the specular (more for small fields of view, less for larger fields of view where diffuse and specular reflection are often both available in the same image). Intuitively, the second direction measures Fresnel effects at grazing angles (large θ_d for fields of view 25° and higher). It can also help refine the diffuse shading, especially for materials with broad specular lobes that cover all of the first image. Figure 3.9 illustrates the one- and two-shot light-view pairs for field of view 25°.

3.7 Results: Near-Field BRDF Measurements

In this section, we apply the reconstruction method, and optimal sampling directions (Sec. 3.6, Fig. 3.8), to image-based near-field BRDF acquisitions of several real samples captured at two different laboratories located in different continents (UCSD in USA and DTU in Denmark). The two laboratory setups deviate slightly and will be described next. We used both approaches to demonstrate the robustness of our method with a variety of simple capture scenarios, which do not require exact configuration or precise alignment between views. We used a portion of the input sample with field of view of 25°, since that achieves near-optimal results (Fig. 3.6).

Setup A (DTU): In this setup, we utilize a high-precision robot-vision system to precisely position the camera relative to a material sample (Fig. 3.10 left). The angular error of this positioning is less than 1 degree. The camera used is a Point Grey Grasshopper 3, industrial CCD camera, mounted with a Kowa LM16SC 16mm lens. The light-source consists of arc holding



(a) Setup A **(b)** Setup B

Figure 3.10. Photograph of our acquisition setup A and B. In setup A, a 6-axis industrial robot precisely positions the camera, and an illumination-arc positioned at $\phi = 0^\circ$ illuminates the sample with halogen lights in 7.5° θ_d intervals. In setup B, a high-angular-resolution spherical gantry positions the light. A DSLR camera is positioned by utilizing the two arms of the gantry and a mirror.

halogen light-bulbs, evenly distributed from 0° to 90° in 7.5° steps.

Setup B (UCSD): We use a (distant) Dolan-Jenner DC 950 light source mounted on one arm of a spherical gantry; the gantry controls two high-precision arms having an angular resolution of 0.1°. However, the gantry's viewing arm/camera is too far away to obtain near-field images directly. We instead manually position a Canon EOS 5D Mark III camera mounted on a tripod close to the input sample (Fig. 3.10 right). In order to correctly position the camera, we place the gantry's two arms at the mirrored direction of the desired viewing location, and adjust the camera until it points towards both arms' center through a mirror. The final position of the camera is obtained through camera-calibration using a checkerboard.

The setups presented above have different limitations, in that the light-source confines setup A to a limited set of view/illumination configurations, whereas the manual positioning of the camera limits the precision of setup B. In both cases, we find the configuration that best matches the optimum directions in Figs. 3.8, 3.9. We thus demonstrate that our method is robust towards small variations in view/light configurations, while still obtaining very good results.

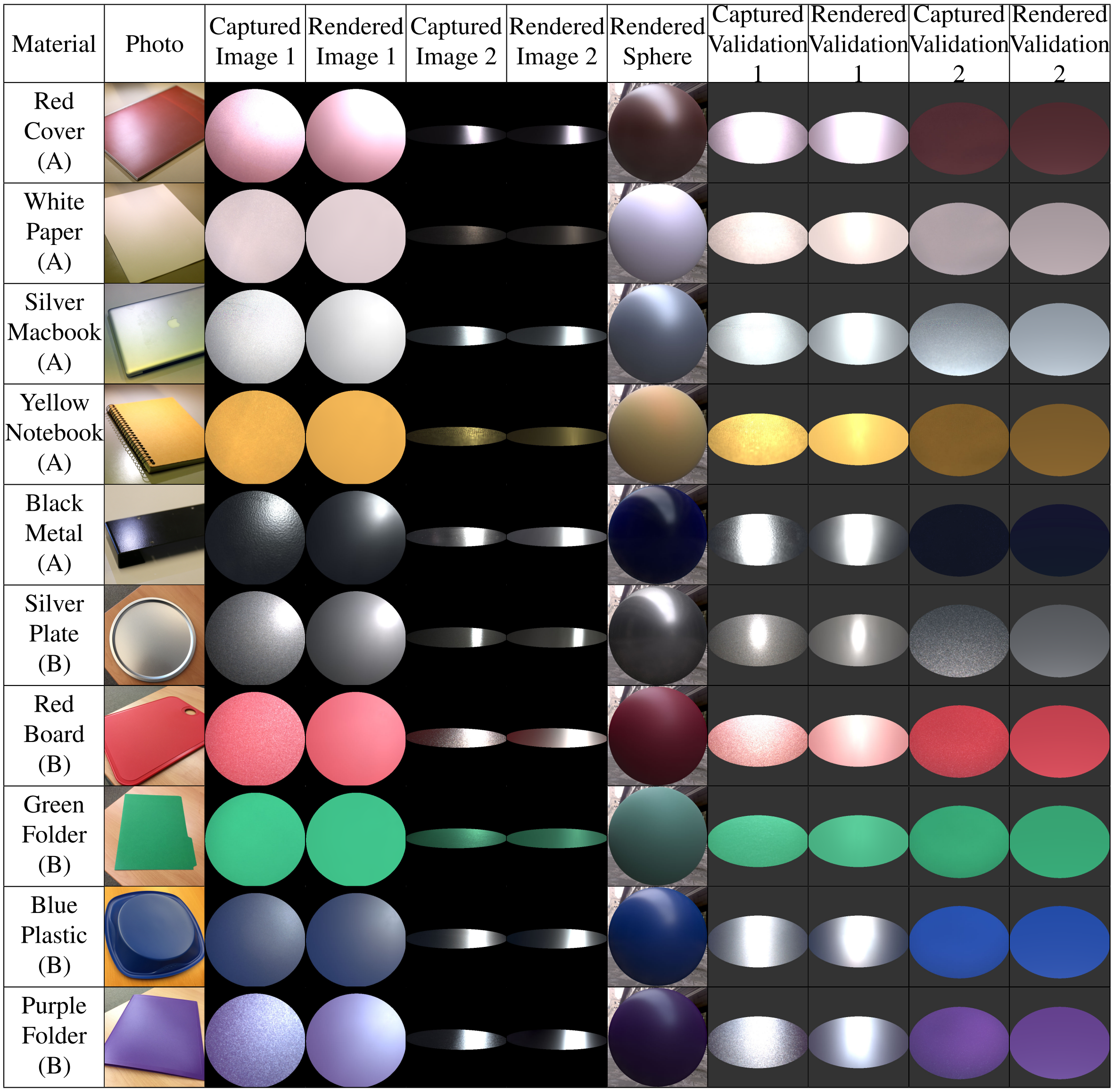


Figure 3.11. Results for near-field BRDF acquisition on real materials. Materials labeled with (A) and (B) are captured by setups A and B. The material is shown first, followed by comparisons of input photographs and renderings with the measured BRDFs. These include both original views, as well as two new lightings and views not used as input. All renderings use the BRDF reconstructed from the two captured images. Good accuracy is obtained for all materials. We also visualize the full BRDF by rendering a sphere lit by an environment map. Note that we use optimal directions in Fig. 3.8 from the full MERL data, which differ slightly from those using 90 materials in Fig. 3.2.

For acquisition, we capture multiple exposures to produce high-dynamic range images; each exposure is averaged over multiple images to reduce noise. The resulting values are then

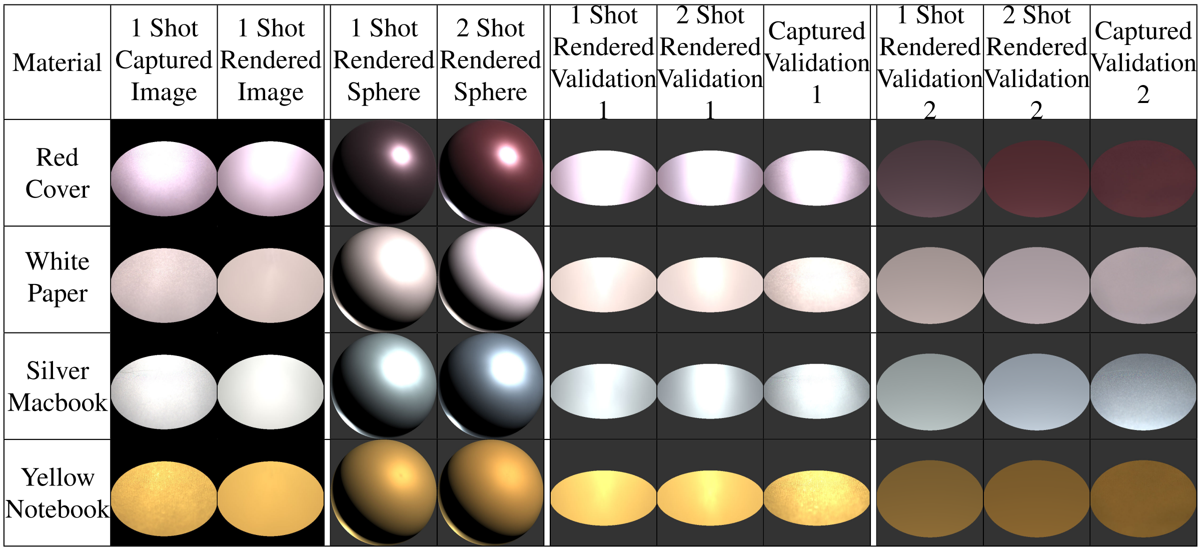


Figure 3.12. Results for single image near-field BRDF acquisition on real materials, comparing one and two-shot reconstructions.

log-mapped, since our framework works with log-mapped BRDFs. Light intensity is calibrated using a Spectralon material sample. By proper modeling of the intrinsic and extrinsic scene-parameters, all observed pixels (BRDF values) can be combined to form $\tilde{x}$ or $\tilde{y} = \tilde{x} - \tilde{\mu}$, using their corresponding view/light coordinates. With this, BRDF coefficients c can be solved from equation 3.4, and the full BRDF x recovered using equation 3.2, with a final step involving undoing the logarithmic mapping. We emphasize that all results in this section were obtained from only two image configurations of a standard flat planar sample, and in some cases only a single image.

Figure 3.11 shows several real samples, whose BRDFs we measured using two-shot near-field acquisition (analogous to Fig. 3.2 for MERL simulations). The first 5 rows are captured using setup A, and the last 5 rows with setup B. The colors differ slightly between the inset photograph and the comparisons, because of the off-white color of the actual light source. We compare the captured image (considering only the circular region of interest, per Fig. 3.4) with the rendering for both light-view input configurations. We also show two additional light-view configurations for validation, which were not used at all as an input. The validation configurations are chosen with $(\theta_h, \theta_d, \phi_d) = (0^\circ, 60^\circ, 0^\circ)\,;\,(22.5^\circ, 22.5^\circ, 0^\circ)$ to verify both specular and diffuse appearance. For visualization, we also show a rendered sphere with the corresponding BRDF lit by an environment map. The accompanying video shows the red cover, white paper and silver macbook under changing viewing directions with two illumination directions, comparing real and rendered results, including fading out the illumination to observe highlights without saturation. It can be seen that the real and rendered images match well. Even when the actual material has some noise or a slightly bumpy surface, we recover a smooth BRDF that is an accurate representation, for both diffuse and more glossy materials.

Finally, Fig. 3.12 shows what can be achieved with a single-shot capture (using setup A). In many cases, one input image is adequate to achieve reasonable results. However, the second input image does help refine the specular reflection somewhat, when comparing the rendered spheres. For example, white paper and yellow notebook are largely diffuse in the first image, and accurate specular and Fresnel information is only achieved at grazing angles in the second view. Moreover, in some cases, the diffuse color and shading can be somewhat refined by using both images.

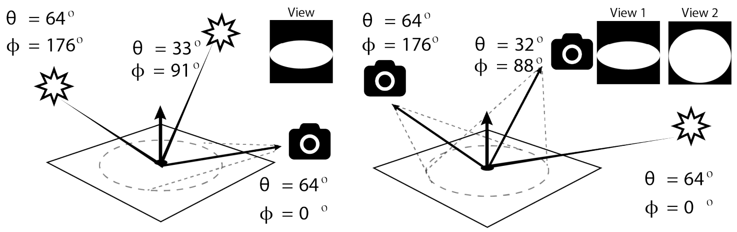
We briefly discuss some limitations. As with all reconstruction methods based on the MERL data, we are ultimately limited by the subspace spanned by that data. Our simulations and experiments indicate excellent agreement with reference measurements, but there is unavoidable error when the material deviates from the MERL subspace. Moreover, for small field of view and BRDFs with broad highlights, our measurements may not capture the full range of the highlight in a single image, leading to under-estimating its width in reconstruction (light red paint in Fig. 3.2). For very dark materials, the noise can be over-fit, causing a blue tint for the black metal in the fifth row of Fig. 3.11. Finally, we do not account for surface imperfections or normal maps, which also contribute to the "noise" above. Nevertheless, as seen in Fig. 3.11, we produce accurate smooth BRDFs consistent with the input data.

3.8 Extension: Fixed Camera Setup

Our error analysis framework and optimization method for sampling directions is general, and could be applied in future to many different configurations. In this section, we consider a one-camera multiple-light case, where we use a single near-field viewing direction, while enabling multiple lighting directions. As before, we show that good results are achieved with two-shot acquisition with two lights. Note that we optimize for lights, and the single view direction, but do so while constraining the camera view to be the same (fixed) for all lighting directions. Using a fixed camera view may also enable simpler acquisition hardware in future. We also briefly discuss the symmetric case of a single fixed light direction, with multiple views. We consider a field of view of 25°.

Figure 3.14 shows the error of fixed camera, multiple light, and fixed light, multiple view, as a function of the number of images, also comparing to our point-sampling and near-field results. As before, our error analysis framework is essential for finding optimal directions, and condition number does not yield meaningful results.

The errors for one or two images are significantly lower than for point-sampling, and only somewhat more than the unconstrained near-field case considered previously (note that one



$\theta_h[°]$	$\theta_d[°]$	$\phi_d[°]$
4	63	90
39	34	124

(a) fixed view 25°

$\theta_h[°]$	$\theta_d[°]$	$\phi_d[°]$
4	63	90
39	33	126

(b) fixed light 25°

Figure 3.13. Optimal configurations and angles for single view, multiple light and single light, multiple view cases, analogous to Fig. 3.9. Angles are shown as standard in-out and Rusinkiewicz coordinates.

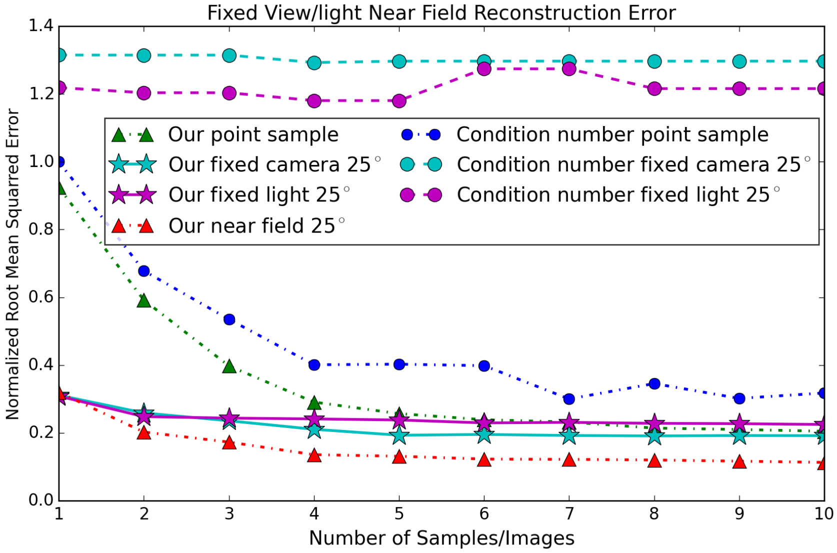


Figure 3.14. Average RMS error over unknown samples for fixed camera/multiple lights, and fixed lighting direction/multiple views.

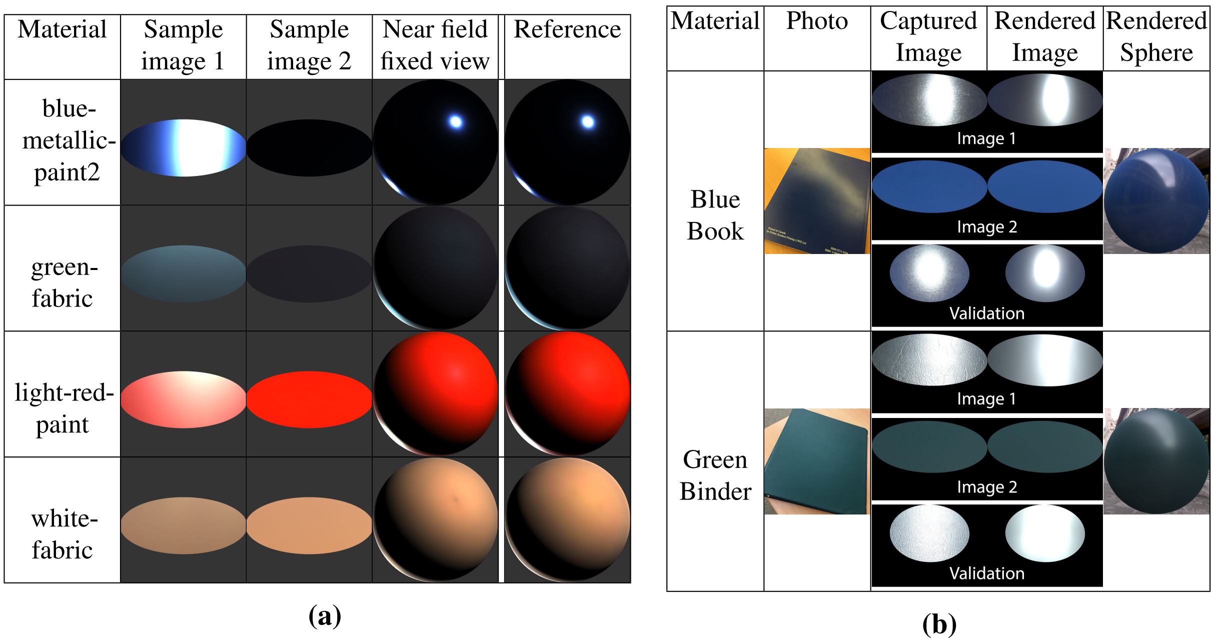


Figure 3.15. Qualitative results for fixed camera acquisition. On the left (a), we show MERL BRDFs reconstructed with fixed view and two images; the configuration works well for most materials. On the right (b), we show acquisition of real BRDFs with fixed camera and two images; we also show a validation view not used as input.

shot acquisition is the same for fixed or free camera setup). However, the lack of flexibility when fixing light or view, leads to a slower decrease in error for more images. Figure 3.13(a) indicates the optimal two-image configuration for fixed camera and changing light. The camera is at a 64° angle to the surface, with light sources positioned to enable observation of both diffuse (light close to zenith) and specular reflectance (light close to mirror direction). Having the camera at an angle to the surface enables capture of some Fresnel information, but a fixed camera setup will make it harder to fully reproduce grazing angles. We also show the analogous configuration for fixed light with multiple views in Fig. 3.13(b). Since fixed light/fixed view configurations are symmetric with similar error, we focus on the fixed camera setup, with only a single viewing direction, and therefore simpler calibration and alignment.

Figure 3.15a shows some synthetic MERL materials reconstructed with fixed view and two images. Note that the first input image is mostly specular while the second is mostly diffuse (dark for specular materials like metallic paint). The results are generally good in most cases. However, some ringing can be observed on the mostly diffuse white fabric. This corresponds to the higher error in Fig. 3.14, compared to the near-field case where both light and camera can move. Figure 3.15b shows comparable results for two real materials captured with setup B. We also show a validation view (specular with light/camera at 45°) not used as input. Good results are achieved with the two-shot fixed camera setup, although there is minor variance in the shape of the specular highlight.

Simple Extension to Spatially-Varying BRDFs: So far, we have not considered spatial variation. We take a first step with a simple extension for specific objects, which have two or more materials that have good coverage over the field of view (such as stars spread out on a background). The fixed-camera setup is ideal for this purpose, since no alignment/calibration between different near-field views is required. Note that this is an initial effort, and further work is required to extend the method to general SVBRDFs.

If we can cluster which pixels correspond to which material, we can separately estimate the BRDFs of the materials, using only the subset of pixels for that BRDF. The key requirement is *coverage* over the field of view, to enable one to see the full range of viewing angles. Using

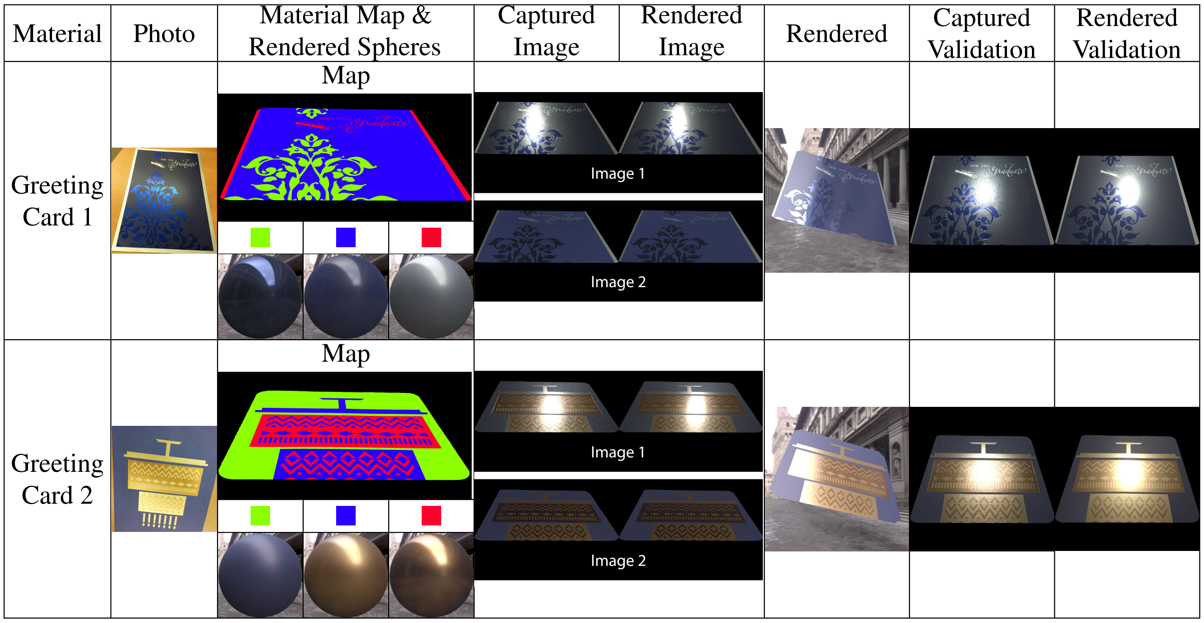


Figure 3.16. Acquisition of spatially-varying BRDFs from two images with fixed camera. Note the close match of captured and rendered images, including in the validation view, not used as input (rightmost column). Rendered images are under environment lighting. We also show the 3 material clusters used in each case, and spheres rendered with the full BRDFs recovered for each of the 3 materials.

only a subset of pixels does not significantly increase error, especially since a 2D image already contains thousands to millions of observations. In practice, we cluster based on color observed in the second (diffuse) captured image. BRDFs are then estimated separately for each cluster. Figure 3.16 shows results for two greeting cards with spatial variation, acquired using setup B. In this example, we consider the full field of view, rather than only a circular region. As seen in Fig. 3.16, we cluster into three materials, and recover full BRDFs for all three materials. The rendered images are close to the captured, with the expected smoothing of surface roughness. (Microstructure and normal variations in the real object cause glints and rough specularities, which increase the apparent size of the highlight for the real object). The validation view, not used as input, also matches well.

3.9 Summary

We have developed a method for acquisition of a full measured isotropic 3D BRDF from only two perspective images of a flat sample, lit with a directional light source. This is at least an order of magnitude reduction in effort over previous comparable techniques to measure a full BRDF, and requires only a standard flat homogeneous material sample. Our method is based on using the full 2D image information from a near-field view, and finds the best lighting and viewing directions by minimizing an estimate of the reconstruction error. We provide tables of these directions for different fields of view of the sample, which can directly be used by other researchers. Our major technical contribution is a formal derivation of reconstruction error, which provides a framework for minimization for both point-sampled and near-field BRDF acquisition, producing better results than the previous condition number heuristic.

In future work, we would like to explore other implications of our method. The new reconstruction error framework could have broad impact in problems like many light methods or computation of light transport matrices, where one seeks to reconstruct from a sparse set of samples. Finally, the one or two-shot nature of our method opens the possibility of designing new simple hardware, with light sources and camera in fixed position.

This chapter is a reformatted version of the material as it appears in "Minimal BRDF Sampling for Two-Shot Near-Field Reflectance Acquisition," Zexiang Xu, Jannik Boll Nielsen, Jiyang, Yu, Henrik Wann Jensen, Ravi Ramamoorthi, ACM Transactions on Graphics (TOG) 35 (6), 2016 [171]. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Deep Image-Based Relighting

4.1 Introduction

We have presented highly efficient approaches for geometry capture in Chapter 2 and material capture in Chapter 3. Once the complete geometry and materials in a scene are fully modeled, synthesizing new images of the scene under new lighting or viewpoint can be done by rendering the reconstruction under the new conditions. However, it is still extremely challenging to do such complete reconstruction for highly complex real scenes with complex geometry and materials, which often leads to image synthesis results that are still easily distinguishable from real images. In this and the next chapter, we study image-based appearance acquisition techniques, which avoid the complicated geometry and reflectance reconstruction, and, instead, focus on photo-realistic image synthesis.

In this chapter, we focus on the problem of rendering a scene under novel lighting, which is a long-studied vision and graphics problem with applications in visual effects, virtual and augmented reality, and product visualization for e-commerce. Image-based relighting methods bypass explicit scene reconstruction by directly modeling the scene's light transport function. Assuming distant illumination, the light transport function, $\mathbf{T}(\mathbf{x}, \boldsymbol{\omega})$, maps incident illumination from direction $\boldsymbol{\omega}$ to outgoing radiance at pixel x (towards the camera), and allows for the scene

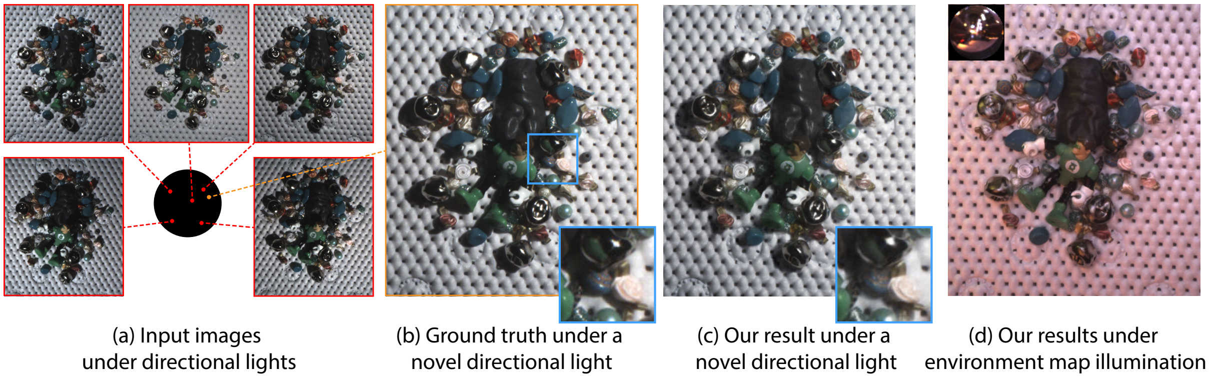

(a) Input images under directional lights
(b) Ground truth under a novel directional light
(c) Our result under a novel directional light
(d) Our results under environment map illumination

Figure 4.1. We propose a learning-based method that takes only five images of a scene under directional lights (a, light directions marked on circle in red) and reconstructs its appearance (c) under a novel directional light in the upper hemisphere (marked in orange). Our method trains a fully-convolutional neural network to jointly learn the optimal input light directions and relighting function for any scene. The network can reconstruct even high-frequency patterns like specular shading and cast shadows (insets in c) and produces photorealistic relighting results that closely match the ground truth (b). Moreover, by generating images for every direction in the upper hemisphere, our method can relight scenes under environment map illumination (d).

to be rendered under novel distant lighting as:

$$\mathbf{I}(\mathbf{x}) = \int_{\mathscr{L}} \mathbf{T}(\mathbf{x}, \boldsymbol{\omega}) L(\boldsymbol{\omega}) d\boldsymbol{\omega}, \tag{4.1}$$

where $L(\boldsymbol{\omega})$ is the radiance of the incident illumination from direction $\boldsymbol{\omega}$. The light transport function can be sampled by capturing images under different lighting conditions; for example, an image of the scene under a single directional light from direction $\boldsymbol{\omega}_j$, yields the sample: $\mathbf{I}_j(\cdot) = \mathbf{T}(\cdot, \boldsymbol{\omega}_j)$. Image-based relighting methods use a set of such samples, $\{(\mathbf{I}_j, \boldsymbol{\omega}_j) \mid j = 1, 2, ..., k\}$, to reproduce scene appearance, $\mathbf{I}_n$, under a novel light, $\boldsymbol{\omega}_n$. Because the light transport function already combines all the interactions of incident illumination with scene geometry and materials, these methods can reproduce photorealistic lighting effects that are difficult to reconstruct and render.

Brute-force image-based relighting methods [31] densely sample the light transport function by capturing hundreds to thousands of images of a scene; they can then relight the scene by interpolating these dense samples. However, the light transport function is known to be highly coherent [139, 106, 96], and this has been used to reconstruct it from a smaller number of images [154, 120] and relight images using lower-dimensional functions [97, 122]. However, these methods still require tens to hundreds of images; this in turn requires considerable acquisition time, and often, specialized hardware.

In this work, we present a technique to render scene appearance under novel illumination from only five images. Previous image-based relighting methods have exploited coherence in a *single* light transport function. Instead, we leverage commonalities between *different* light transport functions, and estimate a single, non-linear, high-dimensional function that maps the appearance of *any* scene under a *sparse* set of pre-defined directional lights to the appearance of that scene under *any* directional light (in the upper hemisphere). Inspired by the success of deep learning at challenging appearance analysis tasks [121, 72, 48], we represent this function using a deep convolutional neural network (Sec. 4.3.1). We train this network — that we refer to as Relight-Net — using a large, synthetically rendered dataset consisting of scenes with procedurally generated shapes and real-world BRDFs (Sec. 4.3.3). Given five images of a scene under directional lights, Relight-Net can reproduce scene appearance under any directional light lying in the visible hemisphere.

The visual quality of Relight-Net's output is a function of the input directions used. Therefore, we design Sample-Net, a custom layer that chooses a sparse subset of a dense set of images. We prepend Sample-Net to Relight-Net to construct an end-to-end network that we train to *jointly learn the optimal input lighting directions and the relighting function* (Sec. 4.3.2). We present an extensive evaluation of our method, including an empirical analysis of reconstruction quality, optimal lighting configurations for different ranges of incident illumination, and alternative network architectures (Sec 4.4.1). We also propose a refinement method that makes Relight-Net robust to small deviations from the optimal input light directions that are likely to occur in real-world capture scenarios (Sec. 4.4.2).

As shown in Figs. 4.1, 4.12 and 4.15, our method generates photorealistic results for real scenes with complex high-frequency effects like cast shadows and sharp specularities. The

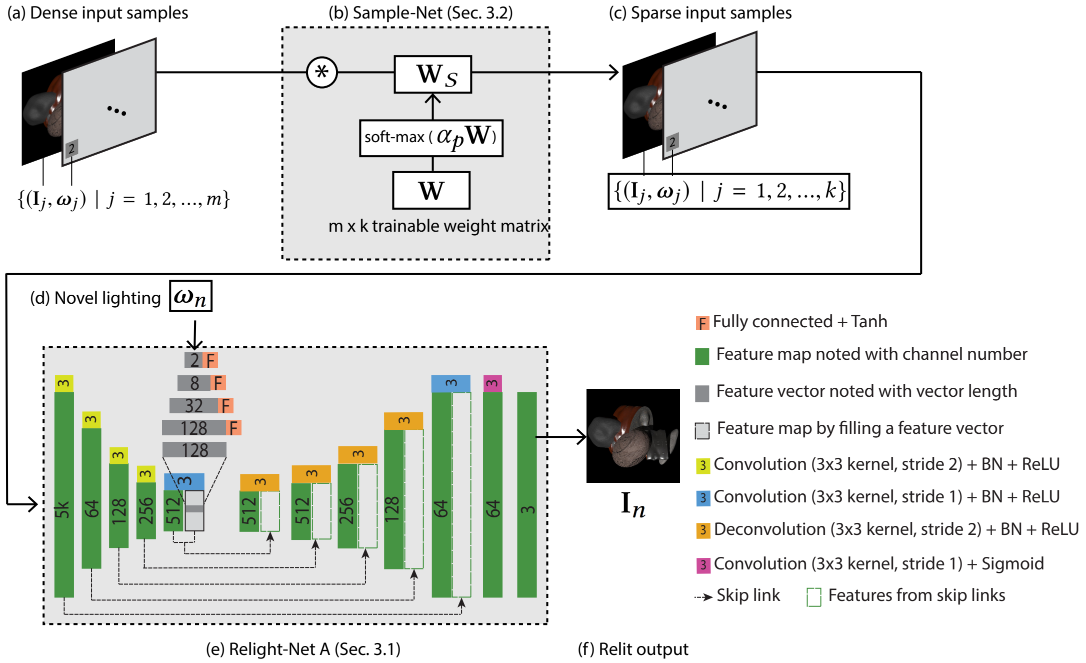

(a) Dense input samples
2
$\{(I_j, \omega_j) \mid j = 1, 2, ..., m\}$
(b) Sample-Net (Sec. 3.2)
$\circledast$
$\mathbf{W}_S$
soft-max ($\alpha_p \mathbf{W}$)
$\mathbf{W}$
m x k trainable weight matrix
(c) Sparse input samples
2
$\{(I_j, \omega_j) \mid j = 1, 2, ..., k\}$
(d) Novel lighting ω_n
2 F
8 F
32 F
128 F
128
3
5k
3
64
3
128
3
256
3
512
3
512
3
512
3
256
3
128
3
64
3
64
3
$\mathbf{I}_n$
F Fully connected + Tanh
Feature map noted with channel number
Feature vector noted with vector length
Feature map by filling a feature vector
3 Convolution (3x3 kernel, stride 2) + BN + ReLU
3 Convolution (3x3 kernel, stride 1) + BN + ReLU
3 Deconvolution (3x3 kernel, stride 2) + BN + ReLU
3 Convolution (3x3 kernel, stride 1) + Sigmoid
Skip link
Features from skip links
(e) Relight-Net A (Sec. 3.1)
(f) Relit output

Figure 4.2. An overview of our network. We stack a dense set of m input images and light directions (a) into a $5m$-channel input that is passed to Sample-Net (b, Sec. 4.3.2). Sample-Net consists of a trainable weight matrix, $\mathbf{W}$, that is multiplied by temperature parameter, α_p and passed through a softmax layer. This constructs a sparse sampling matrix, $\mathbf{W}_s$, that multiplies the dense input to produce the sparse $5k$-channel sample set (c) that is the input to Relight-Net A (e, Sec. 4.3.1). Relight-Net A is a fully-convolutional encoder-decoder; the encoder downsamples the input samples to an intermediate representation. We pass the output light direction, $\boldsymbol{\omega}_n$, (d) through fully-connected layers and replicate and concatenate it to the intermediate representation. The decoder upsamples this to recover the output relit image (f). We use skip links to introduce high-frequency features into the output. We train Sample-Net and Relight-Net jointly to learn both the optimal samples and the relighting function for relighting any scene (Sec. 5.4.6). At test time, we only use Relight-Net to relight the input sparse samples (c) under the input novel lighting (d).

visual quality of our results — generated from just five images — is better than those from previous image-based relighting methods that require an order of magnitude more images (see Figs. 4.12 and 4.13). Thus, our method significantly reduces the acquisition time and complexity for image-based relighting methods and takes a step towards making them more practical.

4.2 Related Work

Dimensionality of Light Transport While changes in illumination can lead to drastic changes in the images of a scene, previous work has shown that these images often lie in low-dimensional subspaces. For example, images of a Lambertian scene are known to lie on a low-dimensional manifold [135, 119, 7, 9, 143]. Even scenes with complex geometry and reflectance have been shown to have low-dimensional light transport in local regions [96], a fact that has been exploited for fast rendering [139, 107] and relighting [106]. These techniques use linear analysis (globally or in local regions) to show the low dimensionality of light transport for a single scene. By exploiting correlations in light transport across scenes using a non-linear CNN-based representation, our work dramatically reduces the number of images required for scene relighting.

Relighting from Sparse Samples While brute-force image-based relighting methods densely sample the light transport function [31], recent methods have leveraged the coherence of the light transport function to reconstruct it using fewer samples. One such approach is to use specially designed illumination patterns during capture [99, 115, 116, 120]. Other methods reconstruct the light transport matrix from a sampled subset of rows or columns [42, 154]. However, these methods still require hundreds of images, and special acquisition systems to create the desired illumination. In contrast, our method can relight scenes from only five images under directional lighting.

Polynomial texture maps (PTM) [97] model per-pixel radiance values as polynomial functions of lighting directions. These functions are fit to (approximately 50) captured images and used to render the scene under novel lighting. Ren et al. [122] use a similar scheme, with the difference that they use shallow neural networks instead of polynomials. They demonstrate impressive results for scenes with complex light transport, but require hundreds of images to achieve this. In contrast, we exploit spatial and angular coherence in light transport across scenes, and learn a more complex, non-linear relighting function to achieve image relighting with only five samples. Figure 4.12 shows that our results have better visual quality than PTM run on 60 images.

BRDF estimation techniques reconstruct BRDFs from images captured under varying illumination. Nielsen et al. [110] and Xu et al. [171] use a linear data-driven BRDF model to derive optimal light directions for BRDF estimation from sparse samples. We propose a technique to learn optimal lighting directions for high-quality scene relighting with a non-linear CNN-based reconstruction method.

Photometric Stereo-based Scene Reconstruction Our acquisition setup is similar to Photometric Stereo methods which reconstruct surface geometry (and reflectance) from images of a scene under varying illumination [164]. While recent techniques can handle non-Lambertian BRDFs [112, 17, 54, 64] they either assume homogeneous BRDFs or require hundreds of image to reconstruct shape and spatially-varying BRDFs. In addition, Photometric Stereo methods often do not consider cast shadows, global illumination, and other light transport effects. In contrast, our method is able to reproduce these effects from fewer samples and outperforms Photometric Stereo-based methods (see Fig. 4.12).

Deep Learning for Appearance Analysis and Synthesis Recently, deep learning-based methods have been successfully applied to inverse rendering and scene reconstruction problems such as reflectance map and illumination estimation [121, 61, 48, 51], reflectance capture [89, 93], and depth and normal estimation [35, 5]. These methods make simplifying assumptions about the scene (for example, considering only a single object) to make the reconstruction tractable. We bypass reconstruction, and directly generate relit images for complex scenes. Deep networks have also been used for view interpolation for relatively unstructured cameras [40] and light field cameras [72]. Our work assumes a fixed viewpoint and attempts to interpolate/extrapolate lighting.

4.3 Learning Image-based Relighting

Given a small set of images of a scene under individual light sources, we want to render the scene under novel lighting. We assume that the scene is imaged from a fixed viewpoint and that the illumination is distant. We also assume that illumination from behind the scene makes a minimal contribution to appearance and can be ignored. Under these assumptions, the light transport matrix, $\mathbf{T}(\mathbf{x}_i, \boldsymbol{\omega}_j)$, represents the proportion of incident radiance from direction, $\boldsymbol{\omega}_j$ (sampled from the upper hemisphere, $\mathscr{L}$) that reaches pixel $\mathbf{x}_i$. Images of the scene under single directional lights represent column-wise samples of the light transport matrix, i.e., $\mathbf{I}_j = \mathbf{T}(:, \boldsymbol{\omega}_j)$.

Given a set of k such samples — images of the scene, $\mathbf{I}_1, \mathbf{I}_2, ...\mathbf{I}_k$, captured under predefined directional lights, $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ..., \boldsymbol{\omega}_k$ respectively — the goal of our work is to reconstruct the image, $\mathbf{I}_n$, that would be produced by a novel directional light, $\boldsymbol{\omega}_n$, via a relighting function, $\Phi(\cdot)$:

$$\mathbf{I}_n = \Phi(\boldsymbol{\omega}_n; \mathbf{I}_1, \boldsymbol{\omega}_1; \mathbf{I}_2, \boldsymbol{\omega}_2; ...; \mathbf{I}_k, \boldsymbol{\omega}_k) = \Phi(\boldsymbol{\omega}_n, \mathbf{S}_1, \mathbf{S}_1, ..., \mathbf{S}_k). \tag{4.2}$$

We hypothesize that $\Phi(\cdot)$ is *scene-agnostic* and can transform sparse input samples, $\mathscr{S} = \{\mathbf{S}_j\} = \{(\mathbf{I}_j, \boldsymbol{\omega}_j)\}$, of *any* scene into a rendering of that scene in novel lighting. We believe this is possible because light transport is highly coherent; our formulation enables this by allowing the radiance at a pixel to potentially be a function of the entire scene under all the sparse light directions.

We model the relighting function, $\Phi(\cdot)$, as a deep convolutional neural network (CNN) that we refer to as **Relight-Net**. We train Relight-Net using a large synthetic dataset consisting of procedurally generated shapes rendered with complex spatially-varying BRDFs, and demonstrate that it can reconstruct high-frequency light transport effects like specularities and cast shadows. Relight-Net is illustrated in the right half of Fig. 4.2 and described in Sec. 4.3.1.

The quality of the reconstruction from Relight-Net depends on the pre-defined lighting directions that we use as input samples. Intuitively, the ability to generalize to new lighting will be limited if the input directions all lie very close together, and will improve as they span the full incident hemisphere. Therefore, we also propose a scheme to learn the optimal input sample directions that lead to the best relighting results. Specifically, we densely sample the space of input light directions, and design a layer that selects a sparse set of these directions. We call this layer **Sample-Net** and describe it in Sec. 4.3.2 (also see left half of Fig. 4.2). We prepend Sample-Net to Relight-Net to construct an end-to-end network that is trained jointly to estimate both the optimal light directions and the corresponding relighting function.

4.3.1 Learning to Relight: Relight-Net

At the core of our method is Relight-Net, a deep fully-convolutional neural network that approximates Eqn. 4.2. We explore two architectures for Relight-Net — the first is a conventional encoder-decoder architecture, while the second disentangles the direct and global illumination components of the scene.

Relight-Net A Our first network architecture, Relight-Net A is designed to directly generate a relit image from sparse input samples, $\mathscr{S} = \{(\mathbf{I}_j, \boldsymbol{\omega}_j) \mid j = 1, ..., k\}$. To pass the input light directions $\boldsymbol{\omega}_j = (s_j, t_j)$ (2D coordinates of the direction vector projected to the $z = 0$ disk) to the network, we construct 2-channel constant images with the same resolution as the input images and s and t in each channel respectively. Concatenating this to the input RGB image yields a 5-channel input per-sample; stacking the k samples leads to a $5k$-channel input to Relight-Net A.

As illustrated in Fig. 4.2, Relight-Net A uses a U-net-style encoder-decoder architecture [128]; the encoder takes the $5k$-channel input, passes it through a series of convolutional layers (with stride 2 for downsampling), each followed by batch normalization (BN) and ReLU layers. The target lighting direction, $\boldsymbol{\omega}_n$, is passed through fully-connected layers (with tanh activation layers after each linear operation) to expand the 2-dimensional vector $\boldsymbol{\omega} = (s, t)$ into a 128-dimensional feature vector. We replicate this feature vector spatially to construct a 128-

channel feature map that is concatenated with the encoder output. The decoder convolves the concatenated encoder output and upsamples the features with deconvolution (transpose convolution) layers, where both convolution and deconvolution are followed by BN and ReLU layers. We use skip connections from the encoder to the decoder to improve per-pixel details in the output. The decoder ends with 2 convolution layers followed by a sigmoid activation to produce the relit image. We train the network using an L_2 loss on the output images, $L_A = \|\mathbf{I}_n - \mathbf{I}_n^{gt}\|_2$, where $\mathbf{I}_n^{gt}$ is the ground truth image rendered with a directional light source at $\boldsymbol{\omega}_n$.

The structure of Relight-Net A allows it to leverage two forms of coherence in a transport matrix: the convolutional layers exploit spatial coherence by aggregating over the network's receptive field, and combining feature maps across channels exploits correlations over lighting directions. This allows it to handle diffuse and specular reflectance, shadowing and other global illumination effects.

Relight-Net B Relight-Net A is designed to directly regress a relit image from input samples, and does not have any rendering-specific constraints. We design an alternative architecture — Relight-Net B — to evaluate if the explicit inclusion of rendering priors can improve the relighting results. Specifically, we know that the appearance of a scene under a single directional light, $\boldsymbol{\omega}_n$, can be represented as a sum of direct and global illumination components: $\mathbf{I}_n = \mathbf{I}_n^d * \mathbf{V}_n + \mathbf{I}_n^g$, where $\mathbf{I}_n^d$, $\mathbf{V}_n$, and $\mathbf{I}_n^g$ are the direct component, per-pixel visibility map w.r.t. $\boldsymbol{\omega}_n$, and global illumination components respectively. We train Relight-Net B to explicitly decode each of these components.

As shown in Fig. 4.3, Relight-Net B consists of a single encoder, connected with three separate decoders. The three decoders generate $\mathbf{I}_n^d$, $\mathbf{V}_n$, and $\mathbf{I}_n^g$, which are then combined to reconstruct $\mathbf{I}_n$. The encoder and decoders are identical to those used in Relight-Net A.

Since we use synthetic rendered data to train the network, we can generate ground truth data for direct illumination images, visibility maps, and final relit images and use them as supervision. We render direct component, $\mathbf{I}_n^{d,gt}$, by computing local per-pixel shading without

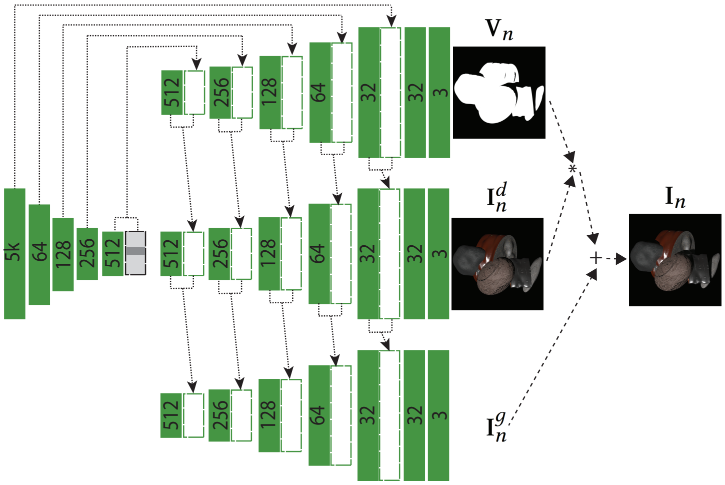


Figure 4.3. Relight-Net B. Similar to Relight-Net A, we use an encoder-decoder architecture. However, Relight-Net B has a single encoder and three separate decoders to reconstruct the direct illumination, visibility map, and indirect illumination images, that are then combined to reconstruct the relit output. We use skip links from the encoder to all the decoders to recover high-frequency details.

considering visibility. We construct the visiblity map, $\mathbf{V}_n^{gt}$, using shadow ray casting. The final loss of Relight-Net B is the sum of three L_2 losses of the three supervised terms: $L_B = \|\mathbf{I}_n^d - \mathbf{I}_n^{d,gt}\|_2 + \|\mathbf{V}_n - \mathbf{V}_n^{gt}\|_2 + \|\mathbf{I}_n - \mathbf{I}_n^{gt}\|_2$.

4.3.2 Learning Optimal Light Samples: Sample-Net

Relight-Net produces relit images from a set of sparse input samples captured under *fixed, predefined* directions, and this form of structured input contributes to the quality of the results. However, the specific choice of directions to use can have a substantial bearing on relighting quality, and it is not clear apriori what the optimal lighting configuration is. One choice for learning the optimal lighting directions is to regress these parameters using the network. However, changes in light direction can lead to complex changes in scene appearance that are challenging to model and are not differentiable w.r.t. to the lighting (e.g., changes in shadows). Instead, we densely sample the domain of incident illumination (i.e., the upper hemisphere, $\mathscr{L}$), pre-render images of the training scenes under these lights, and pose the problem of estimating the optimal samples as one of *selecting* a sparse subset of these dense samples.

Let the dense set of input samples be given by $\mathscr{D} = \{(\mathbf{I}_j, \boldsymbol{\omega}_j) \mid j = 1, ..., m\}$. By vectorizing each $(\mathbf{I}_j, \boldsymbol{\omega}_j)$ pair and stacking these samples, we can construct the $5p \times m$ dense sample matrix $\mathbf{D}$, where p is the number of pixels in the input images. Selecting a subset of these samples can be done as:

$$\mathbf{S} = \mathbf{D}\, \mathbf{W}_S, \tag{4.3}$$

where $\mathbf{W}_S$ is a $m \times k$ binary matrix ($k << m$), where each column has a single 1 entry (corresponding to the sample from $\mathbf{D}$ that is "selected").

Sample-Net is a trainable $m \times k$ $\mathbf{W}$ matrix that post-multiplies the dense input samples to create a sparse set of samples. However, we need to enforce that this matrix is binary and each column only has a single 1. Inspired by Chakrabarti et al. [16], we do this by applying a `softmax` layer to each column of $\mathbf{W}$:

$$\mathbf{W}_S = \text{softmax}(\alpha_p \mathbf{W}), \tag{4.4}$$

where α_p is a scalar parameter that gradually increases from 1 to ∞ during each epoch, p, of the training process. Because of the form of the softmax layer, $\sigma(\mathbf{z})_j = \exp(z_j)/\sum \exp(z_k)$, using a larger α_p makes each column of $\mathbf{W}_S$ sparser. We initialize $\mathbf{W}$ with all 1s. As a result, in the early stages of training, samples in $\mathbf{S}$ are a linear combination of samples in $\mathbf{D}$, but as α_p goes to infinity, each column of $\mathbf{W}_S$ gradually converges to a single non-zero element that corresponds to the chosen optimal sample (see Fig. 4.7 in Section 4.4.1). We use a quadratic model $\alpha_p = \beta p^2$, where β is a tunable hyper parameter.

While Eqn. 4.3 vectorizes each input sample into a $5p \times 1$ vector, in practice we represent them as 5-channel inputs and apply the same value of $\mathbf{W}_s$ to each channel of the sample. Figure 4.2 illustrates how we combine Sample-Net and Relight-Net. In Sec. 5.4.6 we describe how we train them jointly.

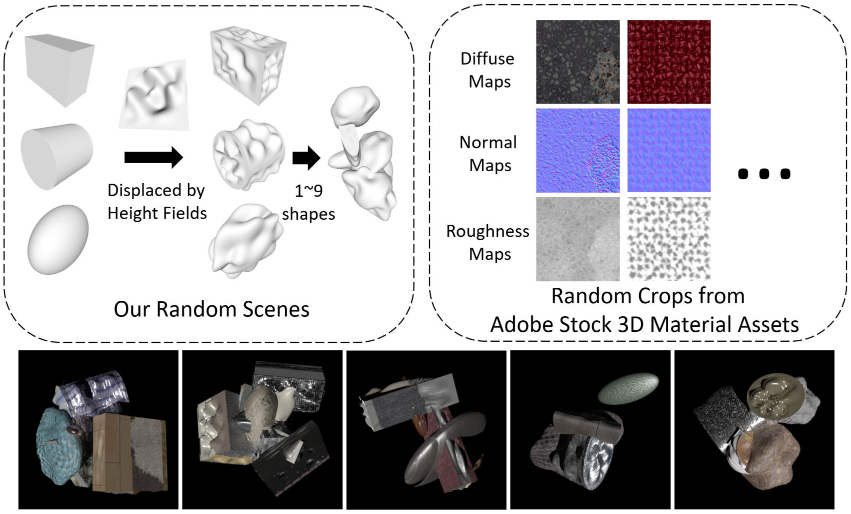


Figure 4.4. Training data. Our scenes consist of multiple $(1-9)$ random primitive shapes that are augmented with varying levels of height fields (top left). We texture these shapes with SVBRDFs from the Adobe Stock 3D Material dataset (top right) and render them using Mitsuba (bottom).

4.3.3 Generating training data

We train our networks with a synthetic rendered dataset; this allows us to control all aspects of our training scenes and also lets us create ground truth data for relit images and intermediate results (like those needed by Relight-Net B).

We generate primitive shapes (cubes, ellipsoids, and cylinders) with random parameters, and apply height-fields with varying frequencies of variation. This gives us a large set of random shapes; we construct the scene geometry by combining multiple (1 to 9) shapes after applying random translations and rotations. We create 600 scenes using this method with 500 for training and 100 for testing.

We use material definitions from the Adobe Stock 3D material dataset[1] — a dataset of 694 realistic spatially-varying BRDFs (SVBRDFs). This dataset uses a physically-based microfacet BRDF model [14] (that uses the GGX distribution [153]) and each material is represented by high resolution (4096×4096) diffuse maps, roughness maps and normal maps. We texture the shapes with random crops from these SVBRDFs (see Fig. 4.4). We separate the SVBRDFs into

[1] https://stock.adobe.com/3d-assets

training set (594 materials) and test set (100 materials); the materials used in the synthetic test scenes are thus never seen during training.

We render 512×512-resolution training and testing images with Mitsuba [70] using bidirectional path tracing [85] with 196 samples. To make our method applicable to low dynamic range images captured by conventional cameras, we apply a gamma of 2.2 and clip the images at 1. We use a combination of Mitsuba's `mixturebsdf`, `roughconductor`, `diffuse`, and `normalmap` plugins to render the materials.

Figure 4.4 illustrates some of our rendered scenes. While the composition of these scenes may not be realistic, note that they locally exhibit the kinds of complex light transport that are present in the real world, including complex surface reflections, cast shadows, and inter-reflections. As we show in Figs. 4.1, 4.12 and 4.15, this allows us to learn a relighting function that generalizes well to real scenes.

4.3.4 Training Relight-Net and Sample-Net

As discussed in Sec. 4.3.2 (and illustrated in Fig. 4.2), Sample-Net is designed to be jointly trained with Relight-Net; given a dense set of scene samples, Sample-Net selects a sparse subset that can be input to Relight-Net to produce the relit result. To train them jointly, we start by densely sampling the incident illumination domain, $\mathscr{L}_\theta$ — a θ-degree cone towards the viewpoint as shown in Fig. 4.5. This gives us a large set of discrete lights $\Omega_\theta = \{\boldsymbol{\omega}_j | j = 1, 2, .., m_\theta\}$. We render each training scene, i, under every light in this set to create the dense input samples $\mathscr{D}_{i,\theta} = \{(\mathbf{I}^{gt}_{i,j}, \boldsymbol{\omega}_j) | \boldsymbol{\omega}_j \in \Omega_\theta\}$. We train the combined Sample-Relight-Net in an end-to-end fashion to minimize the Relight-Net loss function across all scenes and all output light directions:

$$L(\mathbf{W}_s, \Phi) = \sum_i \sum_{\boldsymbol{\omega}_j \in \Omega_\theta} \|\Phi(\boldsymbol{\omega}_j; \mathbf{D}_{i,\theta}\, \mathbf{W}_S) - \mathbf{I}^{gt}_{i,j}\|_2, \tag{4.5}$$

where $\mathbf{D}_{i,\theta}$ is constructed from $\mathscr{D}_{i,\theta}$ as described in Sec. 4.3.2. This loss function evaluates the error of reconstructing images under *every* light in Ω_θ from images under only k lights from Ω_θ.

We crop 10 128×128 patches from each rendered image $\mathbf{I}_{i,j}^{gt}$ giving us 5000 scene-patches for our training. Each training scene-patch has a corresponding $\mathscr{D}_{i,\theta}$. Since training Sample-Net requires loading $\mathscr{D}_{i,\theta}$ completely, we are only able to train with a small batch size. This in turn implies swapping $\mathscr{D}_{i,\theta}$ out repeatedly and can lead to significant I/O overheads. Instead we organize training as follows: in each batch, we load $\mathscr{D}_{i,\theta}$ for 4 random scenes and randomly pick 18 images $(\mathbf{I}_{i,j}^{gt}, \boldsymbol{\omega}_j)$ from each $\mathscr{D}_{i,\theta}$ as targets for Relight-Net to reconstruct. This forms a batch of 4 scenes for Sample-Net training and 72 images for Relight-Net training. We use ADAM with 0.0001 as the learning rate for joint training. β from 5 to 8 generally works well, and we use $\beta = 6$. We find that our networks typically converge after 16 epochs. Our final learned models, scenes, rendered images and the code for generating them are released on the project website.[2]

Since Relight-Net is fully convolutional, at test time we can apply it to arbitrary resolution images, although it only considers appearance within a 128×128 window size. Moreover, while Relight-Net has been trained using only the discrete lights in Ω_θ, we show that it can be used to relight using any directional light on the continuous domain $\mathscr{L}_\theta$.

4.4 Analysis on Synthetic Data

4.4.1 Analysis of Relight-Net and Sample-Net

In this section, we present analysis and empirical evaluations of the different components of our network. Unless otherwise specified, we use the Relight-Net A for testing. Later in the section, we compare Relight-Net A and Relight-Net B.

Light domain vs. number of sparse samples By training Sample-Net and Relight-Net to minimize Eqn. 4.5, we can learn to relight a scene from any light direction in $\mathscr{L}_\theta$. To investigate the effect of the size of this domain (in terms of cone angle, θ) on the performance of our network, we train our network on four light domains: $\theta = \{30, 45, 60, 90\}$. We create

[2] http://viscomp.ucsd.edu/projects/SIG18Relighting

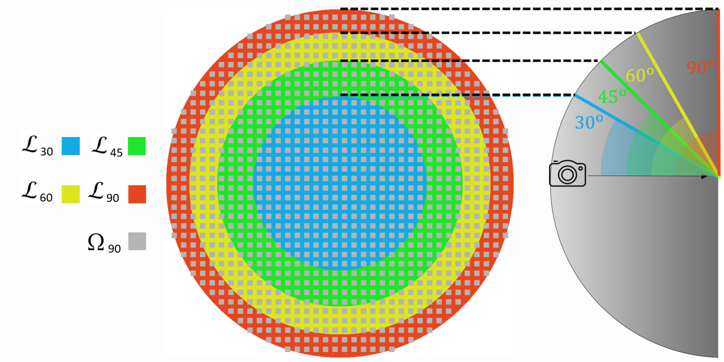

$\mathcal{L}_{30}$ 🟦 $\mathcal{L}_{45}$ 🟩
$\mathcal{L}_{60}$ 🟨 $\mathcal{L}_{90}$ 🟥
Ω_{90} ⬜
30°
45°
60°
90°

Figure 4.5. Incident illumination domains and samples. As shown on the right, we model scene appearance under directional lights that lie on $\mathscr{L}_\theta$, a θ degree cone on the hemisphere pointing towards the viewing direction. On the left we illustrate the hemisphere and different light domains, $\mathscr{L}_{30}$, $\mathscr{L}_{45}$, $\mathscr{L}_{60}$, $\mathscr{L}_{90}$, in a 2D plane. We also show Ω_{90}, the densely sampled 1052 discrete directions from $\mathscr{L}_{90}$. Other Ω_θ are constructed from Ω_{90} as $\Omega_{30} = \Omega_{90} \cap \mathscr{L}_{30}, \Omega_{45} = \Omega_{90} \cap \mathscr{L}_{45}, \Omega_{60} = \Omega_{90} \cap \mathscr{L}_{60}$.

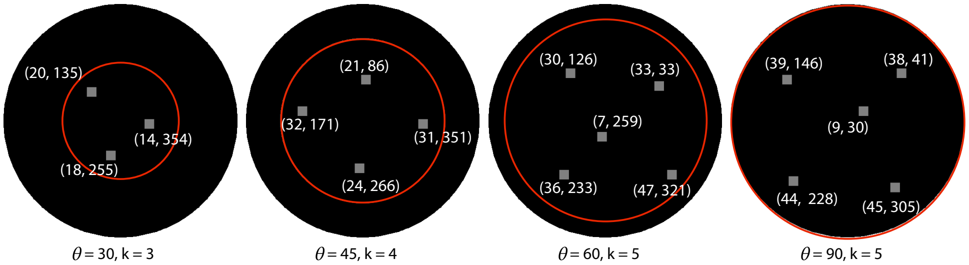


Figure 4.6. Learnt optimal directions for several lighting configurations. We represent directions using the standard (θ, ϕ) spherical parameterization.

Ω_{90} by uniformly sampling 38 values for the (s,t) coordinates of light directions in the domain $(-0.952, 0.952)$ and rejecting samples outside the unit disk. This leads to $m_{90} = 1052$ distinct light directions in $\mathscr{L}_{90}$. Ω_{30}, Ω_{45} and Ω_{60} are subsets of Ω_{90}, with $m_{30} = 256$, $m_{45} = 540$ and $m_{60} = 804$ directions, respectively. Details about $\mathscr{L}_\theta$ and Ω_θ are shown in Fig. 4.5.

In addition to the size of the illumination domain, the quality of our reconstructions depends on the number of sparse samples that are input to Relight-Net. In particular, as the size of the illumination domain increases, we expect that we would need more samples to preserve reconstruction quality. Therefore, we analyze the performance of our full Sample-Net-Relight-Net architecture for 12 different light domain size/sample configurations: $\theta = 30, k = \{2,3,4\}$; $\theta = 45, k = \{4,5\}$; $\theta = 60, k = \{5,6,7\}$; and $\theta = 90, k = \{5,6,7,8\}$.

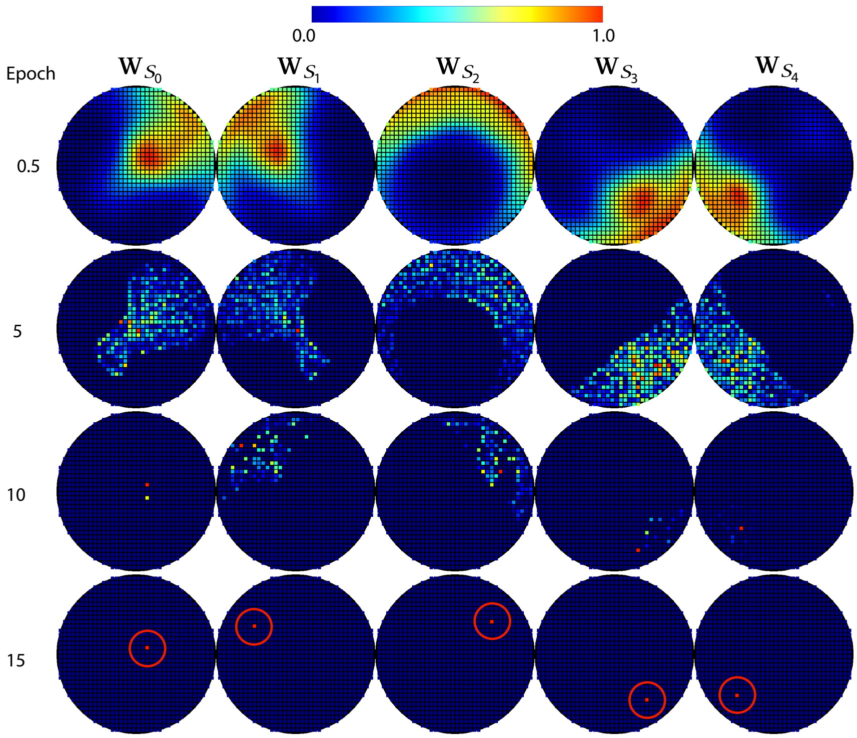


Figure 4.7. Evolution of the optimal sparse samples during joint training for $(\theta = 90, k = 5)$; each column represents the values from one column of $\mathbf{W}_S$. Starting from a flat distribution, each column of $\mathbf{W}_S$ gradually becomes peakier, till it converges to a single sample at epoch 15.

Learnt optimal samples Over the course of our joint training process, Sample-Net gradually converges to k optimal samples. In Fig. 4.7, we illustrate this for $\theta = 90$ and $k = 5$ samples. $\mathbf{W}_S$ starts off as a mixing of many samples and gradually converges to 5 optimal samples. As we would expect intuitively, these samples are distributed over $\mathscr{L}_{90}$.

Figure 4.6 indicates the learnt optimal directions for 4 representative networks, one for each light angle setting. We can see that when $k = 3, 4$, the optimal directions are spread in a circle around the center of the cone; setting $k = 5$ adds a direction near the center of the cone, i.e., nearly collocated with the viewpoint. Also note that all optimal directions are not placed at the edge of lighting domain, indicating that Sample-Net chooses directions that allow Relight-Net to both interpolate and extrapolate the input samples to produce relit results. Note that Sample-Net could have converged to a local minima, as is often the case with deep networks. However, in practice we have found that these directions lead to better reconstructions than arbitrarily chosen

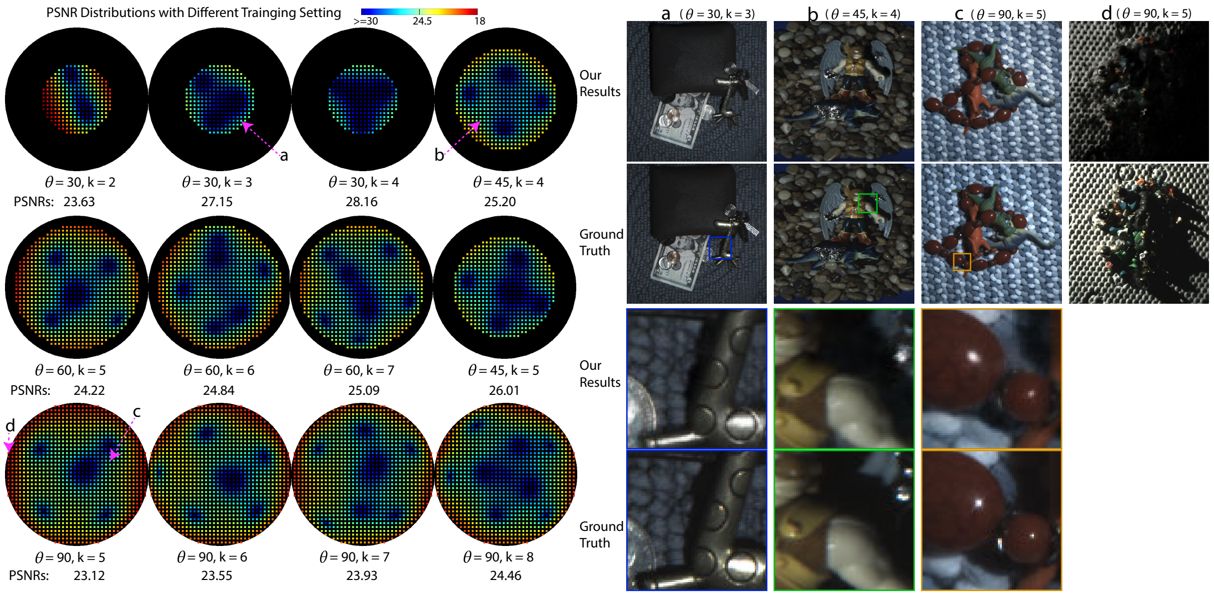


Figure 4.8. Reconstruction errors for different lighting configurations. On the left, we visualize reconstruction quality (in PSNR) for each test light direction aggregated over all our test scenes. Errors are lowest for directions close to the input samples, and are quite low in the convex hull of the input samples even for directions away from input samples. The average PSNR across all light directions is listed under each figure. On the right, we show relighting results for 4 real scenes (a,b,c,d) captured with different light setups and compare them against ground truth. The corresponding output directions are marked on the left. Our reconstructions are very accurate for $(\theta = 30, k = 3)$, $(\theta = 45, k = 4)$, and most light directions in $(\theta = 90, k = 5)$. Note that the $(\theta = 90, k = 5)$ setup can fail for single directional lights at extreme grazing angles (d), but this has minimal impact when integrating over an environment map as seen in Figs. 4.1 and 4.15.

directions and other sampling strategies based on heuristics, as we will discuss shortly. Optimal directions for the remaining 8 (θ, k) configurations are shown in Appendix D.

Reconstruction quality We use our trained networks to relight the 100-scene test set under all the lighting directions in the trained light domain, and aggregate the errors. We perform this analysis for different choices of (θ,k) and illustrate the results in Fig. 4.8. From these error distributions, we can make the following observations about Relight-Net: 1) it produces very low reconstruction errors for lights close to the input samples; 2) it produces high-quality results for interpolated light samples, i.e., output light directions that lie within the convex hull of the input light directions; 3) while it is able to do extrapolate to relight scenes under lights that lie outside the convex hull of the sparse input samples, the errors are larger that those for interpolation.

While each scene might have its own optimal sampling directions based on its geometry and reflectance properties, the directions in Fig. 4.6 are optimal for all scenes. In addition, the optimal directions are chosen to let Relight-Net trade-off errors in interpolation and extrapolation scenarios.

As expected, the reconstruction error is lower for smaller light domains; for example, the network trained for $(\theta = 45, k = 5)$ has a PSNR of 26.01 vs. 23.12 for $(\theta = 90, k = 5)$. Our network with $(\theta = 45, k = 4)$ produces near-photorealistic results for most directions in $\mathscr{L}_{45}$. For $\theta = 90$, i.e., the entire upper hemisphere, the lighting setup we have showcased in this chapter, $(\theta = 90, k = 5)$, produces accurate results across much of the light domain. This network might blur some high-frequency effects like sharp shadows and small specularities. However, these issues are most evident when we render the scene under high-frequency directional lights; rendering the scene under environment map illumination leads to results that are perceptually indistinguishable from ground truth images (see Fig. 4.14). Moreover, using additional samples, e.g., $k = 8$, also improves performance. These experiments suggest that we can use Sample-Net to further optimize the sample configuration for specific scenes or capture scenarios.

Comparisons against alternative sampling strategies To evaluate the quality of our learnt samples, we compare them against heuristics-based strategies that produce well-distributed samples. We choose two methods — random dart throwing and k-means clustering, and evaluate them on the $(\theta = 90, k = 5)$ configuration.

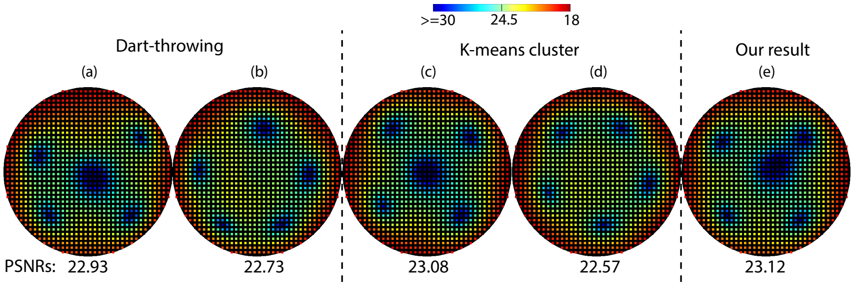


Figure 4.9. Comparison with two random dart throwing sample sets (a,b), two representative k-Means clustering samples (c,d), and our samples (e). Our samples produce higher average PSNR.

To ensure that random dart throwing leads to well-distributed samples, we specify a minimal threshold on the distance between two samples. We also apply the same minimal threshold on the distance from the boundary of the light domain, i.e., the grazing angle. Without this condition, we found that samples tend to converge towards the boundary of the domain (which has more samples) which leads to poor relighting performance. Since we do not know the threshold distance apriori, we generate sample sets with gradually increasing thresholds and pick the value, 40°, which allows only five samples. Our other baseline uses k-means clustering to group the 1052 samples in Ω_{90} into 5 clusters. We found that k-means clustering generally converges to two types of distributions: either one central direction with four directions distributed around it, or five directions around the center of the light domain (see Fig. 4.9(c,d)). We randomly select two results of random dart throwing and two of k-means clustering (from the two representative distributions). We train the Relight-Net by using these samples as the input and using the same 5000 scene-patches $\mathscr{D}_{i,\theta}$ as training data.

We test these trained networks on the 100-scene test and compare the reconstruction error. As shown in Fig. 4.9, our Sample-Net samples significantly outperform the random dart throwing ones (Fig. 4.9 (a,b)). k-means clustering is not reliable either; while the first distribution (Fig. 4.9 (c)) approaches our performance, the second distribution is significantly worse (Fig. 4.9 (d)). Moreover, it is not easy to predict what the relighting performance of any of these sampling strategies would be, without training Relight-Net for each of them. In contrast, using Sample-Net in conjunction with Relight-Net allows us, in a single training pass, to jointly learn the optimal samples and the relighting function that *maximizes relighting performance*.

Relight-Net A vs. Relight-Net B We evaluate the effect of introducing rendering constraints into the relighting function, by comparing the performance of Relight-Net A and Relight-Net B for the same $(\theta = 90, k = 5)$ configuration. For this experiment, we first trained Sample-Net+Relight-Net A to learn the optimal input directions and then trained Relight-Net B with the same directions. Fig. 4.10 shows a comparison of the relighting error for these two networks.

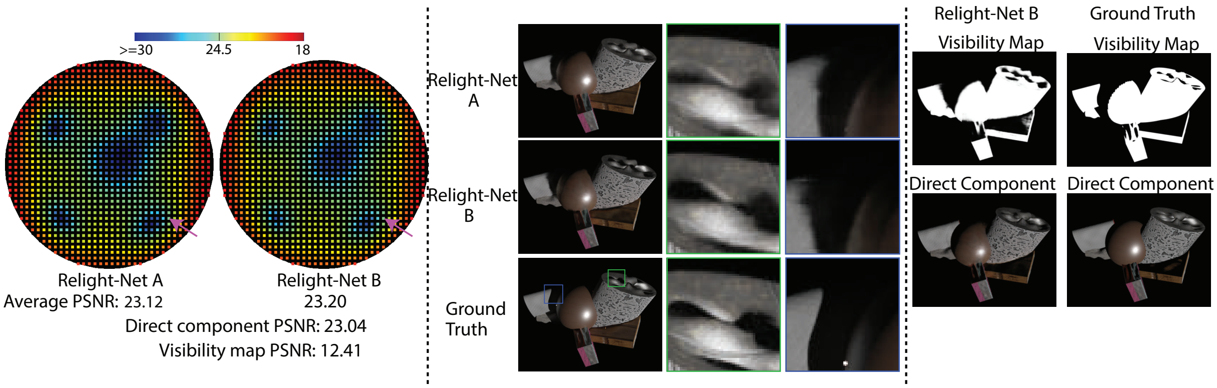


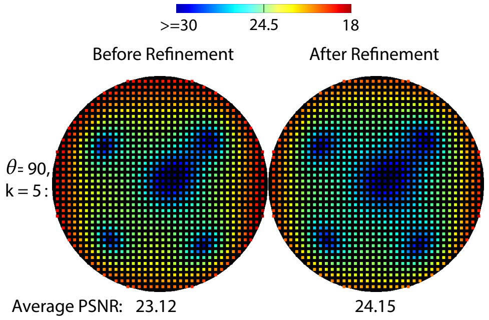
Figure 4.10. Relight-Net A vs. Relight-Net B. We compare relighting errors across all light directions for our two Relight-Net architectures for the $(\theta = 90, k = 5)$ lighting configuration using the same optimal light directions (top left). Overall PSNR and PSNR for each Relight-Net B component are also listed. Relight-Net B has marginally better performance but visual inspection (top right) for one scene and light direction (marked by the arrow on the left), shows that it produces shadows that are better sometimes (green inset) and inaccurate at other times (blue inset). We also show the visibility map and direct component from Relight-Net B (bottom).

While Relight-Net B has marginally better average performance (PSNR of 23.20 vs. 23.12 for Relight-Net A), it does not consistently outperform Relight-Net A in terms of visual quality. This might be attributable to the difficulty of learning the high-frequency visibility function (PSNR of 12.41). In contrast, the reconstruction of the direct illumination component is quite accurate (PSNR of 23.04).

We chose Relight-Net A for all the experiments in this chapter because it matches Relight-Net B's *relighting* performance and is faster to evaluate. However, Relight-Net B's scene decomposition results suggest that using this technique for scene *reconstruction* could be an interesting direction of future work.

4.4.2 Refining Relight-Net

After joint training, Relight-Net can relight a scene from the k sparse samples from the learnt optimal directions. However, this would require an acquisition system to recreate these optimal light directions exactly, which can be challenging in practice. In order to reduce this requirement, we refine Relight-Net by training it to handle input light directions in the local



Average PSNRs for inputs away from the optimal

All	1-5 degree	5-10 degree
23.82	24.02	23.59

Figure 4.11. Evaluation of network refinement. On the left we compare reconstruction errors for optimal inputs for Relight-Net before and after refinement with $(\theta = 90, k = 5)$. Refinement improves the results because it has been trained on a larger dataset. Moreover, the refined network also performs well on two scenarios of non-optimal input directions that are 1–5° and 5–10° off the optimal. As the average PSNR (computed across our entire test set) for these scenarios shows, the refined network is quite robust to these deviations, and in fact outperforms the non-refined network's performance on optimal inputs.

neighborhood of the optimal directions. Note that we are able to do this because the input light directions are one of the inputs to Relight-Net.

To refine Relight-Net, we generate a new training dataset comprised of the original 500 training scenes as well as a new set of 5000 scenes (for a total of 5500 scenes). For each scene, we render a new set of k random input samples (sampled within a 10° cone around the learnt optimal light directions) and another 50 output images under random light directions over the entire $\mathscr{L}_\theta$ cone. These images are generated as described in Sec. 4.3.3. As before, we refine Relight-Net using 10 random 128×128 crops from each image.

Figure 4.11 compares Relight-Net error distributions before and after refinement on the same test dataset. We can see that refinement improves reconstruction quality (23.12 before refinement vs. 24.15 after) even when we use the optimal input directions that the joint optimization selected. This is a consequence of refining Relight-Net on a larger (5500 vs. 500) scene dataset; while we could have trained our combined Sample-Net+Relight-Net on this dataset, the large computational requirements to train Sample-Net make this intractable. More importantly, the refined network is able to handle inputs that are away from the optimal directions. To evaulate this, we randomly select two sets of 10 input directions that are 1–5° and 5–10° away from the optimal directions, and test relighting performance for these directions on our entire test dataset. The average PSNRs of these 2 settings are shown on the right in Fig. 4.11. We can see that even when the inputs vary by 5–10° from the optimal directions, the refined network achieves 23.59 average PSNR, and in fact, outperforms the non-refined network's results with the optimal input samples. As we show in our experiments on real data, this robustness to the input directions allows our method to produce high-quality results even for datasets captured by acquisition setups that do not exactly meet our specifications.

4.5 Results and Evaluation

We now present an evaluation and comparisons of our method on both synthetic and real data. Unless otherwise specified, the results in this section were generated using our refined Relight-Net A model trained for the $(\theta = 90, k = 5)$ setting. We encourage readers to zoom into the images in this section to look at image details.

Datasets We evaluate our method on the synthetic scenes from our 100-scene test data, as well as three different real datasets — one that we captured ourselves using a gantry-based acquisition system (Figs. 4.8 and 4.15), and additional scenes captured with a light stage setup by Einarsson et al. [36] (Figs. 4.12 and 4.15) and Schwartz et al. [132] (Fig. 4.17). While we could control our gantry to capture images under our learnt optimal directions, the latter two datasets do not have these directions, and the closest directions deviate by an average of 4°. Our results for these scenes rely on the refined network's robustness to input light directions.

Timing We can relight a scene under a directional light using a forward pass through our network model; this takes 0.03 seconds on a NVIDIA Geforce 1080Ti for a 512x512-resolution image.

Comparison with Photometric Stereo-based reconstruction As mentioned in Sec. 4.1, one approach to image relighting is to reconstruct the scene and re-render it under novel lighting.

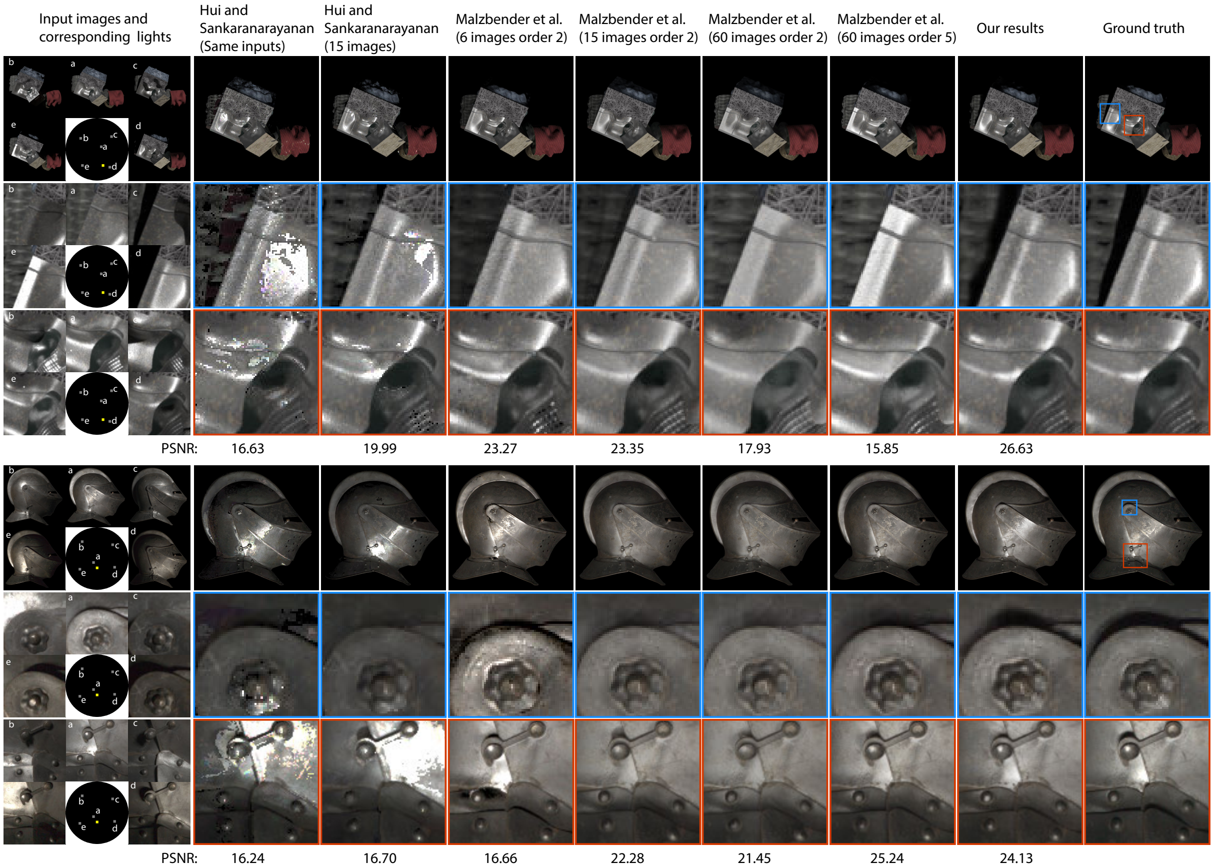

Input images and corresponding lights
Hui and Sankaranarayanan (Same inputs)
Hui and Sankaranarayanan (15 images)
Malzbender et al. (6 images order 2)
Malzbender et al. (15 images order 2)
Malzbender et al. (60 images order 2)
Malzbender et al. (60 images order 5)
Our results
Ground truth
b
a
c
e
d
PSNR: 16.63 19.99 23.27 23.35 17.93 15.85 26.63
PSNR: 16.24 16.70 16.66 22.28 21.45 25.24 24.13

Figure 4.12. Comparisons with Photometric Stereo [64] (with 5 and 15 samples), and Polynomial Texture Maps [97] (with 6, 15, and 60 samples) on a synthetic test scene (top) and a real scene captured by [36] (bottom, the input light directions here deviate from our optimal directions by 3-7°). The input images and corresponding lights are shown on the left and the PSNR of the result is listed below the images. Our results have some of the highest PSNR scores even when compared to methods with more input images. Moreover, our results have better visual quality and reproduce cast shadows and specularities better (insets).

To compare against this approach, we use a state-of-the-art Photometric Stereo method that can handle spatially-varying BRDFs [64]. Fig. 4.12 shows comparisons with this method when run on 5 and 15 images. Even a state-of-the-art Photometric Stereo method has large errors when reconstructing a scene from a small number of images, resulting in significant artifacts in the relit results. Moreover, this method does not handle non-local effects like cast shadows and inter-reflections. In comparison, our results are significantly better in terms of both PSNR and visual quality.

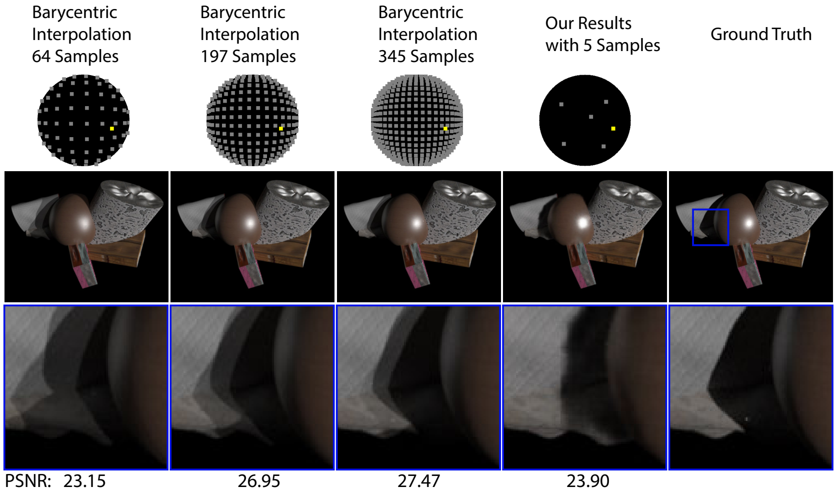

Barycentric Interpolation 64 Samples
Barycentric Interpolation 197 Samples
Barycentric Interpolation 345 Samples
Our Results with 5 Samples
Ground Truth
PSNR: 23.15
26.95
27.47
23.90

Figure 4.13. Comparisons with barycentric interpolation with increasing sampling resolution (top row, samples shown as gray dots and relighting direction in yellow). Our method has a better PSNR than barycentric interpolation with 64 samples. At higher resolutions, barycentric interpolation produces better PSNR, but the subjective visual quality is worse. For example, there are significant ghosting artifacts at shadow boundaries (inset, bottom row).

Comparison with image-based relighting methods Most image-based relighting methods are designed for dense input samples captured using specialized hardware. Therefore, we choose two representative methods —Polynomial Texture Mapping (PTM) [97] and barycentric interpolation — and compare their performance on different sample sets to our results (Fig. 4.12 and Fig. 4.13). For PTM, we fit order-2 polynomials to 6, 15, and 60 input images, and order-5 polynomials to 60 input images. Our network outperforms PTM in most of these settings on both synthetic and real data. Using order-5 polynomials and 60 samples (12× as many as we use) allows PTM to outperform our PSNR on a real scene, but, unlike our network, it can't reconstruct specularities and completely blurs shadows. Moreover, PTM's performance does not consistently improve when we add more samples or use higher-order polynomials. This is possibly because polynomials are poor approximations of light transport, especially in the presence of cast shadows, and using an L_2 error to fit them can lead to unstable results. In general, PTM was designed for largely planar scenes with minimal cast shadows. In contrast, our method can handle more complex scenes with a fraction of the number of samples.

We analyze how many samples are required for barycentric interpolation to approach our result quality in Fig. 4.13. We uniformly sample the upper hemisphere with 64, 197 and 345 directions. We render a synthetic scene at these directions, and use these images to do barycentric interpolation for the frontal hemisphere. As shown in Fig. 4.13, our model produces a reasonable result with plausible (though slightly jagged) shadows even for a novel relighting direction that is outside the convex hull of the input samples. Our method, with 5 samples, has a PSNR of 23.90 vs. 23.15 for barycentric interpolation with 64 images. As the resolution of the sampling is increased, barycentric interpolation starts to outperform our result quantitatively (PSNR of 26.95 and 27.47 for 197 and 345 images). However, the visual quality of our result is still superior; the barycentric interpolation results have significant ghosting artifacts even at 345 input images. These issues are exacerbated in animations with a moving light; our reconstructed shadows and specularities move smoothly and intuitively, while the barycentric interpolation results exhibit significant spatial ghosting and temporal aliasing.

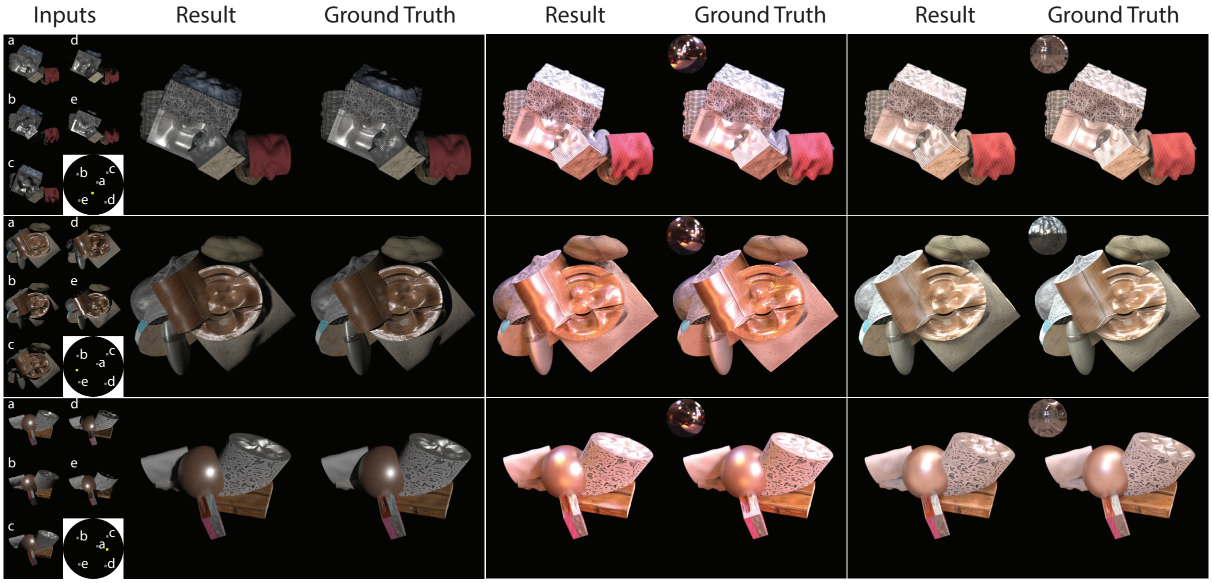


Figure 4.14. Relighting results from five samples for synthetic data (first column, input/output light directions marked on the black circle in gray/yellow). Our results match the ground truth images quite faithfully (left vs. right of second column) with some errors near hard shadow boundaries. Moreover, these errors are visually imperceptible under all-frequency hemispherical environment map illumination (third and fourth columns, environment maps in inset).

Directional and environment map relighting Using a network trained to handle directional lights in $\mathscr{L}_{90}$ allows us to relight a scene under (the upper hemisphere of) environment lighting by rendering images under every direction in the environment map and summing them using weights based on the environment map radiance values. We use a 64×64 hemispherical environment map (3000 output directions). Note that our network design allows us to pre-compute the encoder features for the input images once, and only process the decoder for different light directions.

Figure 4.14 shows a comparison of our results for both directional and environment map illumination with ground truth images for four synthetic test scenes. Note that our results under directional lighting match the ground truth images closely with some minor artifacts along sharp shadow boundaries. Moreover, our results under all-frequency environment lighting — generated from only 5 sparse samples — are visually imperceptible from ground truth results. This indicates that, when rendering under environment illumination, the accuracy of our method at relighting most of the directions in $\mathscr{L}_{90}$ is sufficient to compensate for the errors that occur at grazing angles (as shown in Fig. 4.8).

As Fig. 4.15 demonstrates, we observe similar behavior in real scenes exhibiting a wide range of materials (diffuse to highly specular), geometries (arbitrary shapes arranged in complex ways), and scene scales and layouts (small to medium to large objects). Our network faithfully reproduces appearance under novel directional lights, and creates photorealistic results under environment map illumination. The bottom two results in this figure come from the [36] whose light samples deviate from our exact optimal light directions by 3° to 7°. Yet, we obtain high-quality results illustrating the ability of our network to relight using light directions that are not perfectly optimal.

Limitations While our method produces results of a high quality, some artifacts still remain. While we capture the general shape of cast shadows, the edges can have artifacts (Fig. 4.13). We train our network on 128×128 image patches; this determines the receptive field

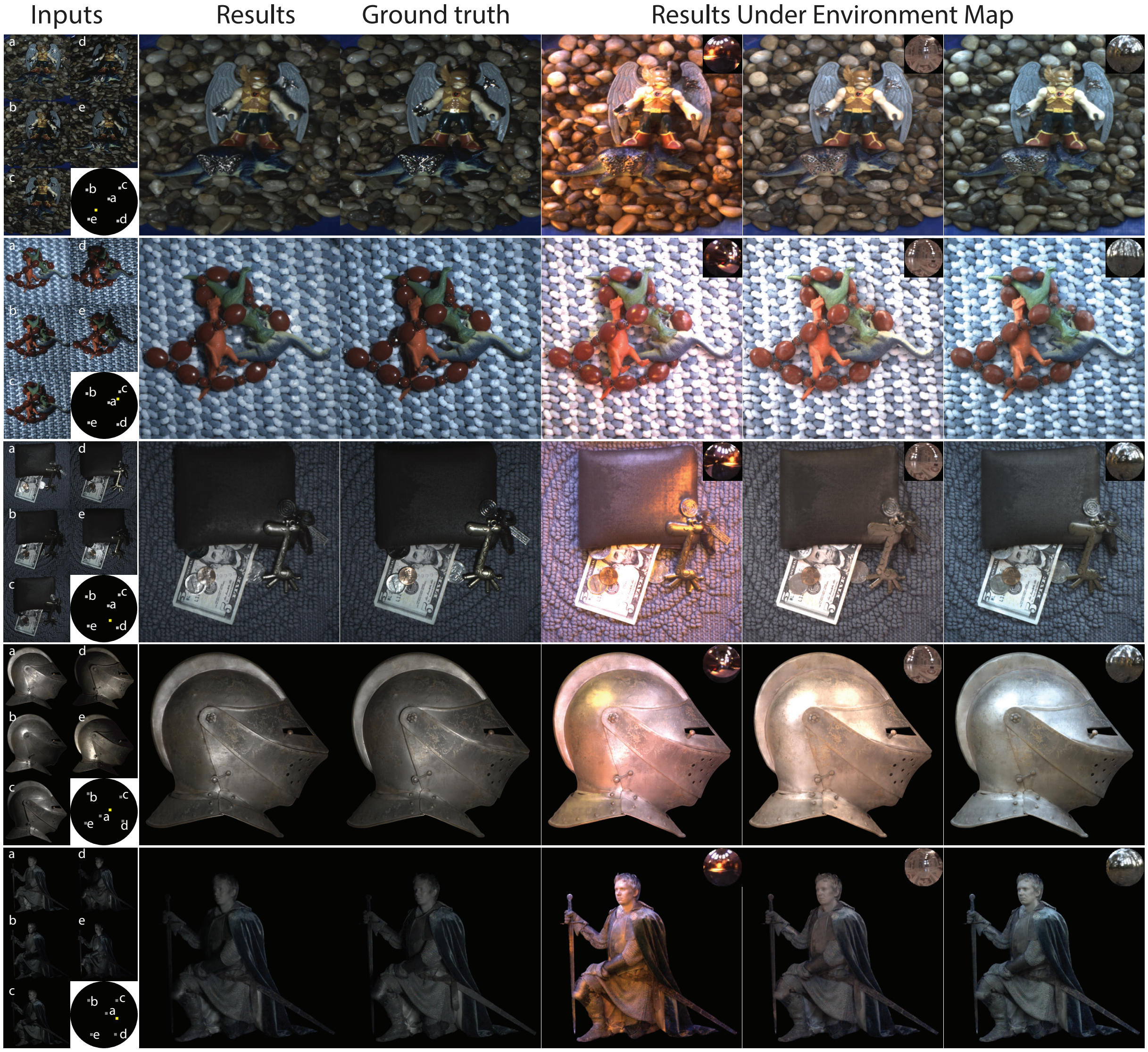


Figure 4.15. Real scenes (top 3 captured by us, bottom 2 from [36]) rendered with environment map lighting. These scenes contain objects with complex reflectances, intricate geometries and span a wide range of scene size and layout. Yet, our method produces accurate relighting results for a single directional light (second column and third column; output direction marked in yellow) and under environment lighting (fourth to sixth columns).

of the features and the spatial scale at which we can analyze scene appearance. Consequently, our method cannot handle non-local appearance changes that happen at a larger scale, for example shadows caused by grazing angle lighting (Fig. 4.8) or highly non-convex geometry (Fig. 4.17). Our method might blur very sharp specularities (Fig. 4.15). As shown in Figs. 4.17 and 4.16, these issues can be ameliorated using more samples or when rendering under environment

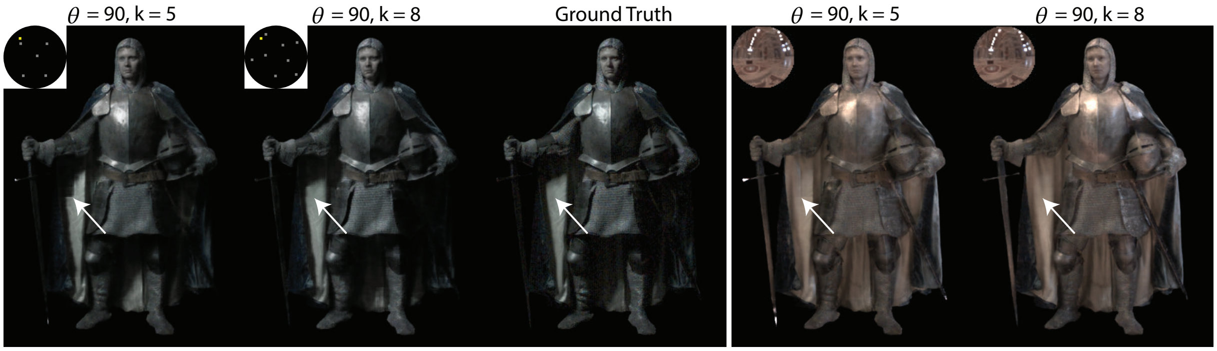


Figure 4.16. Comparison between $k = 5$ vs $k = 8$ for $\theta = 90$. In this case, the incident light causes very significant cast shadows, leading to a shadow artifact with sharp corners (noted by the white arrow) in the $k = 5$ result. The artifact goes away when we use $k = 8$ samples. The artifact is also mitigated under environment lighting (right).

lighting.

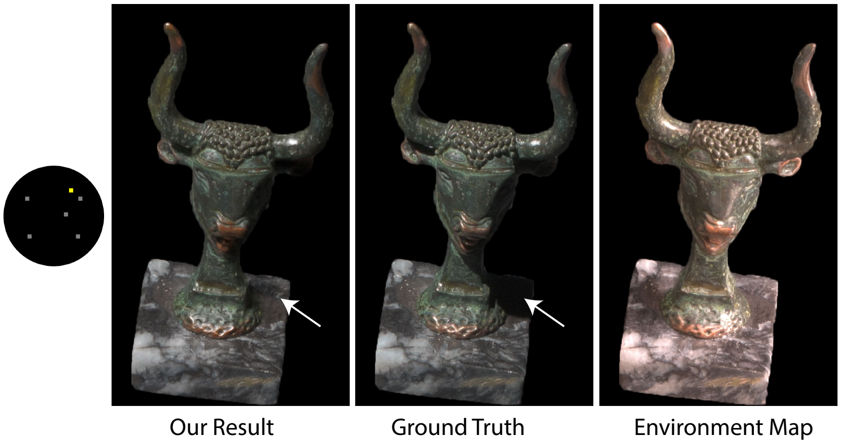


Figure 4.17. Limitations. Our method fails to recover cast shadows caused by highly non-convex geometry. However renderings under environment map illumination are still plausible as shown in the rightmost column.

Our results are also limited by our training data, and the assumptions made to generate it. For example, we assume that objects in the scene are opaque and don't model complex effects like glints. Increasing the diversity of the shapes, materials, and composition of our scenes could help mitigate this.

4.6 Summary

We have presented a novel approach to relighting a scene from a sparse set of input images. We are able to accomplish this by training a CNN to take 5 images of a scene under single directional lights and render the scene under a novel directional light (in the upper hemisphere). Moreover, we present a scheme to learn the optimal directions for these sparse samples in conjunction with the relighting function by jointly training a combined sampling-relighting network. Extensive evaluations and comparisons to previous state-of-the-art image-based relighting approaches show that we are able to achieve the same (if not better) performance as them, except with an order of magnitude fewer input samples.

This chapter suggests a number of interesting directions for future work. At a high-level, most previous scene appearance analysis has relied on simple linear analysis tools. On the other hand, deep networks have been extremely successful at learning good representations for images; can we use them similarly to learn representations for scene appearance? How can we use such representations to reduce the memory and time to relight (or render) a scene? In this work, we learn the optimal set of *directional* lights; how would this change if we also allowed non-directional, *general* illumination? While we have avoided explicit scene reconstruction in this work, the results from training Relight-Net B (Fig. 4.10) indicate that a network could learn to decompose scene factors from input samples. Combining this with learning the lighting that gives the best reconstruction could be another interesting extension.

This chapter is a reformatted version of the material as it appears in “Deep Image-Based Relighting from Optimal Sparse Samples,” Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, Ravi Ramamoorthi, ACM Transactions on Graphics (TOG) 37 (4), 2018 [172]. The dissertation author was the primary investigator and author of this paper.

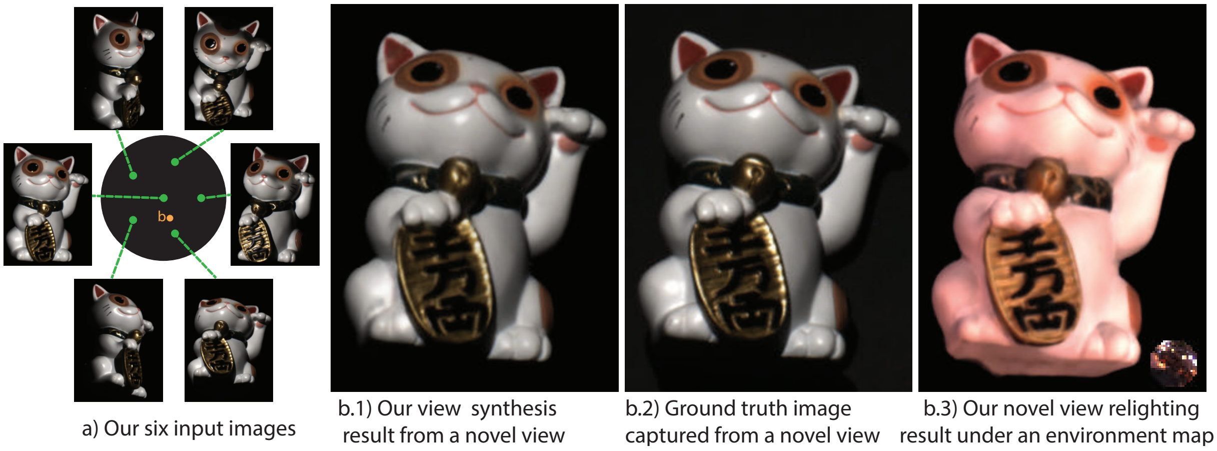
Chapter 5

Deep View Synthesis

5.1 Introduction

In Chapter 4, we have demonstrated that image-based relighting can be achieved from only a sparse set of input images under sparse directional lights. Image-based relighting reproduces scene appearance under varying lighting conditions, in which, however, the viewpoint is always fixed. In this chapter, we explore an orthogonal image-based appearance acquisition problem, novel view synthesis, which aims to render a scene under varying viewpoints. Novel view synthesis methods still require a dense sampling of a scene's "light field" [87, 56]. Consequently, they capture hundreds of images, especially when the scene has complex surface reflectance [163]. While recent work has addressed novel view synthesis from sparse images [72, 40, 140], these methods are highly restricted in the range of viewpoints they can synthesize.

Similar to Chapter 4, we consider "photometric" images – that refer to images captured under controlled directional lighting – as input in view synthesis, which capture complex scene appearance. In this chapter, our goal is to make appearance acquisition and rendering more practical by synthesizing *a wide range of novel viewpoints* from *a sparse set of images*. To do so, we image the scene, consisting of several objects, with six cameras placed on a vertex and the centers of the adjoining faces of a regular icosahedron (see Fig. 5.2). This results in a central camera, and five distributed symmetrically around it at an angle of about 37°. At this large baseline, the captured images have significant occlusions (see Fig. 5.1(b)). State-of-the-art



a) Our six input images

b.1) Our view synthesis result from a novel view

b.2) Ground truth image captured from a novel view

b.3) Our novel view relighting result under an environment map

Figure 5.1. We present a method to synthesize scene appearance from a novel view by interpolating only six wide-baseline images (a). We do this by using a structured setup to capture photometric images under directional lighting and interpolating them using a novel deep neural network. Our method can reproduce complex appearance effects like specularities, shadows, and occlusions (b.1) resulting in images that are close to ground truth captured images (b.2). These results can be combined with our deep image-based relighting methods in chapter 4 to visualize the scene under novel view and lighting (b.3).

multi-view stereo methods fail to reconstruct complete geometry from such sparse views. Yet, we show that our method can interpolate the entire convex hull of these six viewpoints — a cone of more than 60° — while accurately reproducing effects like complex occlusions and high-frequency, view-dependent specularities (see Fig. 5.1(b)).

This is made possible by a combination of our structured acquisition procedure and a novel learning-based interpolation scheme. Unlike previous view synthesis approaches that capture images under environment illumination, we acquire images under a single directional light. These "photometric" images capture appearance information like shading, shadows, and specularities, and have been used for scene reconstruction via Photometric Stereo methods [165] (after which they are named) and image-based relighting [31]. We show that using such inputs leads to view synthesis results that capture detailed scene appearance; this also enables other applications like novel view relighting (see Fig. 5.1, 5.11).

We introduce a novel deep convolutional neural network that learns to interpolate the wide-baseline photometric images and render arbitrary output viewpoints between them. Similar to previous learning-based view synthesis methods [40], our network projects the input images onto multiple depth planes of the output view to construct a view-dependent plane sweep volume. This volume is processed to predict the final output view (and depth) using 3D convolutional layers with downsampling and upsampling operations; this allows the network to reason about both geometry and appearance at multiple spatial scales. A key, novel component of our method is that we explicitly predict *per-plane per-input-view attention maps*. These attention maps are used to modulate the 3D plane sweep volume and allow the network to aggregate multi-view appearance while accounting for occlusions, viewpoint variations, etc. (similar to blending weights in IBR methods), and lead to sharper, more accurate results. We supervise the output image and depth map, and our network learns to predict the attention maps in an unsupervised fashion.

We train our CNN with a large-scale, synthetic dataset consisting of 1000 scenes with procedurally generated shapes and complex spatially-varying BRDFs. We render these scenes with our six-view camera configuration under random directional lighting using path tracing. The rendered images approximate real-world appearance and light transport well, allowing the network to generalize well to real captured scenes. This can be seen in Fig. 5.1, where our method generates photorealistic interpolated results for a real object with complex geometry and appearance effects, including challenging occlusions, high-frequency specularities and cast shadows.

We also demonstrate extensions of our approach that go beyond view synthesis. We can add multiple lights to our acquisition setup to capture a sparse set of multi-view, multi-light images. Our view synthesis network can be extended to synthesize appearance under *novel view and lighting* from this sparse data (see Fig. 5.1b3). We also show that our view synthesis network can be used to "densify" the captured views of a scene; these dense views can then be input to a multi-view stereo algorithm to produce reconstructions that are significantly better than what is possible with just the sparse inputs (see Fig. 5.12). By making these methods work with sparse image sets, our work takes a step towards eliminating the need for dense capture systems and making scene acquisition more practical.

5.2 Related Works

Light transport acquisition. Traditionally, light transport acquisition methods use complex systems to capture images of a scene under a dense set of lighting and view directions. These samples can then be used to explicitly reconstruct scene geometry and reflectance [31, 44, 62, 185, 133]. Recent methods have demonstrated geometry and reflectance reconstruction under more relaxed settings such as handheld capture [105] and unknown environment lighting [169]. For a more detailed discussion of the previous work on appearance acquisition, we refer readers to [161, 158]. While these methods can produce high-quality rendering results, they still require capturing hundreds of images.

This requirement can be relaxed for specific applications. For example, image-based relighting methods focus on directly combining images captured under varying lighting conditions to relight the scene under novel lighting. While earlier image-based relighting methods required brute-force sampling [31, 97], recent methods have leveraged light transport coherence to reduce the number of images required [116]. In particular, Xu et al. [172] demonstrate image-based relighting from only five images. Sparse capture has also been shown to be sufficient for reflectance estimation for planar samples [171, 90, 65], scenes with known geometry [186] or single-view reflectance [6, 91]. Our work focuses on capturing multi-view scene appearance and can reproduce complex effects like view-dependent specularities and occlusions from only six images. Moreover, our method can be extended to capture scene appearance under both varying view and lighting from only 36 images. This is a significant change from traditional light transport acquisition methods that require hundreds to thousands of images.

Novel view synthesis. Novel view synthesis methods focus on interpolating the appearance of the scene between captured images [23], i.e., the view interpolation aspect of the light transport acquisition problem. This can be done by re-sampling rays from a densely captured light field [56, 87]; such light fields can be also reconstructed from sparse samples by leveraging various forms of correspondence between multiple views [152, 175, 26]. Alternatively, image-based rendering methods project captured images onto proxy geometry (that can be given or reconstructed from the captured images), blend and resample them from the novel viewpoint [32, 13, 138, 20, 19].

Penner and Zhang [117] propose soft 3D, which reconstructs depths and visibilities for each input view to achieve better blending. Other works present different blending techniques to achieve ghosting free synthesis results for inaccurate 3D reconstruction [37, 181, 10]. Hedman et al. [58] present a method that learns to predict the blending weights. These blending techniques require scanned or MVS-reconstructed geometry as an input, for which densely sampled viewpoints are necessary. All these methods rely on capturing tens to hundreds of images with large overlap and reasonable baseline; without this, the geometric reconstruction and view synthesis would fail. Moreover, these methods are usually designed for free-viewpoint navigation and do not focus on reproducing detailed appearance like sharp specularities. In contrast, our method works with only six images that are captured with a fairly large baseline, and reproduces complex scene appearance accurately. It does so in an end-to-end fashion and learns to predict appearance, depths and attention/blending maps.

Surface light fields [163] are designed to capture complex scene appearance under high-frequency point lighting. These methods reconstruct the surface geometry and represent appearance on the surface using lumispheres. Because of the high-dimensionality of this representation, methods have focused on compressing this data using PCA and vector quantization [163] or deep networks [21]. Our work is able to capture similar appearance effects with vastly fewer images. Moreover, our network can be thought of as a scene-agnostic representation that can interpolate any scene's images to a new viewpoint.

Learning-based novel view synthesis. Recently, deep learning techniques have been applied to novel view synthesis to achieve unstructured multi-view interpolation [40], narrow-baseline stereo extrapolation [183], narrow-baseline interpolation [72] and single-view extrapolation [140] in the context of light fields. All these methods estimate geometry; either a single depth [72, 140] or per-plane depth probabilities (or blending weights) [40, 183] are predicted for either one input view [183, 140] or each novel view [40, 72]. All these methods resolve visibilities in an implicit way, whereas our network learns multi-view correspondence and explicitly predicts per-input-view visibility-aware attention maps jointly with depth probability maps at a novel viewpoint. Most importantly, we demonstrate that we can synthesize a much wider range of new viewpoints compared to these works.

Geometry-free learning-based methods can directly generate pixels for a novel view from one or multiple input views [174, 145], though these methods are restricted to specific shape classes. Other methods leverage flow prediction to warp pixels from source views [184, 113]. Flow-based warping has been combined with per-input-view confidence maps to improve aggregation [141]. While we also use attention maps, ours are predicted from geometric correspondences, inferred jointly with depths, and incorporate information about occlusions, viewpoint differences, etc. We show that this leads to results that are more realistic than flow-based methods.

Learning-based appearance acquisition. Deep learning-based methods have been applied to appearance acquisition applications like reflectance capture [89], reflectance map estimation [121], and depth estimation [34]. Using photometric images has been shown to be better for single-shot BRDF acquisition [89, 91, 33], and image-based relighting from sparse samples [172]. All these methods focus on the single-view setting. In this work, we explore multi-view appearance acquisition using photometric measurements and achieve photorealistic novel view synthesis under a single directional light from six sparse views.

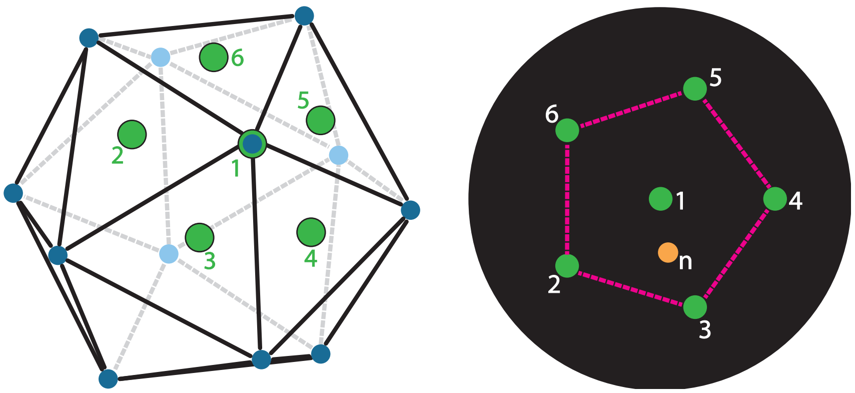


Figure 5.2. Acquisition configuration. Left: a regular convex icosahedron with 12 spherically symmetric vertices. Right: a projective view of our configuration, where the black background circle represents a hemisphere towards the central view. We consider a setup with six known views (green circles, denoted by numbers 1-6), in which one is located at a vertex and five at the adjoining face centers. Our goal is to synthesize a novel view (orange circle, noted by n), in the convex hull of the six known views (red dash-lines).

5.3 Acquisition Configuration

Similar to previous work on reflectance field acquisiton [31, 162], we choose an acquisition setup where cameras are placed on a sphere. We assume that the cameras are distant with respect to the scale of the scene and image the scene with a field-of-view that ensures sufficient pixel coverage. Our goal is to acquire the appearance of a generic scene, without making any assumptions about scene composition. Therefore we utilize a symmetric camera configuration, that would be optimal on average. Given these design decisions, we need to choose a configuration that symmetrically samples a sphere. This sampling has to balance two constraints: we would like a sparse sampling that minimizes acquisition cost while ensuring that views have sufficient overlap to allow high-quality view interpolation.

Our configuration of choice, as shown in Fig. 5.2, is inspired by a standard spherically symmetric shape: the regular convex icosahedron. An icosahedron has 12 vertices with 20 equilateral triangular faces, symmetrically distributed around a sphere. Given an icosahedron, we investigate a setup with $m = 6$ cameras, in which one camera is positioned on a vertex and five "boundary" cameras are positioned at the centers of the faces that surround the central vertex.

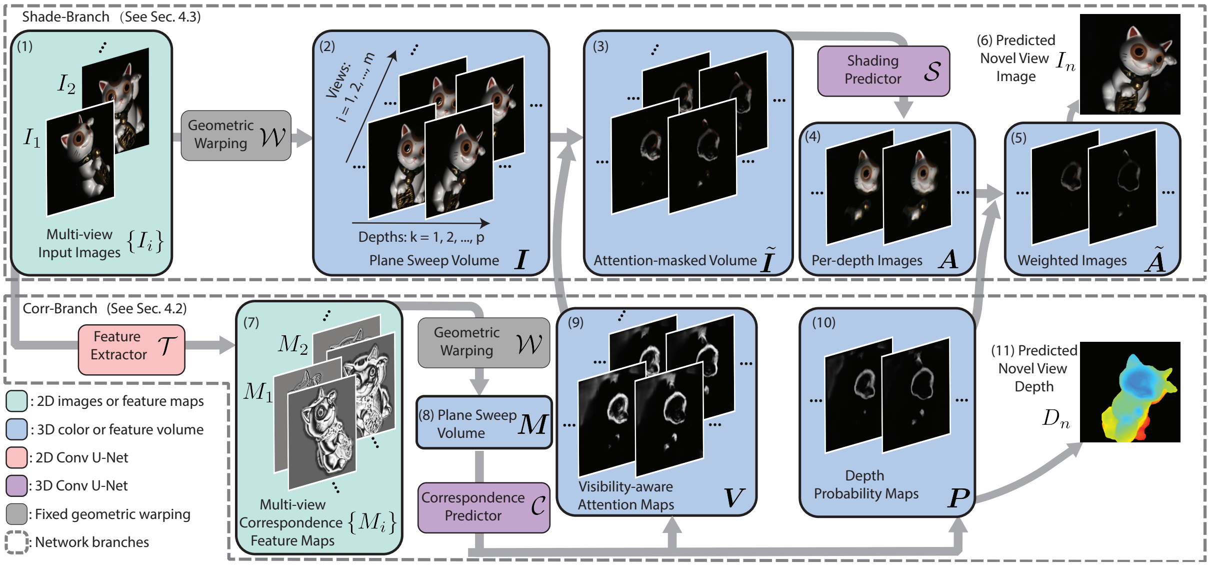

Shade-Branch (See Sec. 4.3)
(1)
I_2
I_1
Multi-view Input Images $\{I_i\}$
Geometric Warping $\mathcal{W}$
(2)
Views: i = 1, 2, ..., m
Depths: k = 1, 2, ..., p
Plane Sweep Volume $\boldsymbol{I}$
(3)
Attention-masked Volume $\tilde{\boldsymbol{I}}$
Shading Predictor $\mathcal{S}$
(4)
Per-depth Images $\boldsymbol{A}$
(6) Predicted Novel View Image I_n
(5)
Weighted Images $\tilde{\boldsymbol{A}}$
Corr-Branch (See Sec. 4.2)
Feature Extractor $\mathcal{T}$
(7)
M_2
M_1
Multi-view Correspondence Feature Maps $\{M_i\}$
Geometric Warping $\mathcal{W}$
(8) Plane Sweep Volume $\boldsymbol{M}$
Correspondence Predictor $\mathcal{C}$
(9)
Visibility-aware Attention Maps $\boldsymbol{V}$
(10)
Depth Probability Maps $\boldsymbol{P}$
(11) Predicted Novel View Depth
D_n
: 2D images or feature maps
: 3D color or feature volume
: 2D Conv U-Net
: 3D Conv U-Net
: Fixed geometric warping
: Network branches

Figure 5.3. Network Overview. Our view synthesis network consists of two branches that operate on plane sweep volumes (shown in blue) that are constructed using geometric warping (gray boxes). **Corr-Branch** (bottom) extracts image features (7) and estimates attention maps (9) and depth probability maps (10). The depth probabilites are used to estimate scene depth from the novel view (11). **Shade-Branch** (top) processes the input image volume using the attention maps and depth probabilities to synthesize the novel view image (6). Please refer to Fig. 5.4 and 5.5 for details of $\mathcal{T}$, $\mathcal{C}$ and $\mathcal{S}$.

In this configuration, the angular distance from the central view to each boundary view is about 37°. Note that this is a very wide baseline, and our cameras observe very different parts of a scene with limited overlapping regions. Our goal is to synthesize an arbitrary view point in the convex hull of these six known views; this represents a cone with an angular baseline of more than 60°. Because of the symmetry of the icosahedron, our six-view setup can potentially be extended for full sphere acquisition. Cameras can be placed at every vertex and every face center, in total requiring merely 32 views. This is a very sparse camera setup with significantly fewer views compared to previous techniques [163, 21] that capture hundreds of images.

We focus on a practical case where a scene is composed of one or a few objects that are placed on a flat platform. We capture photometric images of this scene lit by an arbitrary directional light from the frontal hemisphere (the black circular region in Fig. 5.2 right). This region is most likely to illuminate the scene and light coming from behind the scene will have little contribution. For our multi-view multi-light extension (Sec. 5.4.4), we use six fixed lights, with one collocated with the central camera, and five located on the surrounding neighboring five vertices. Fixing the light directions in this scenario allows the network to leverage this structured input and produce higher quality synthesis results.

5.4 Algorithm

Given images from our six pre-defined, calibrated views under a single directional light, our goal is to synthesize a new image from a specified novel view between the six inputs. Inspired by the recent success of deep learning, we propose to train a deep CNN to directly regress the final output image. Our method is a geometry-based IBR method that uses scene geometry to align and blend the multi-view data. However, instead of relying on a precomputed geometric proxy [13, 58], we design our network to infer multi-view correspondence information for each novel view, and predict novel view depths as side products. Since we are using a synthetic dataset, the ground truth for both final images and view-dependent depths can be generated and used as supervision. Our network leverages this supervision to learn shading and correspondence simultaneously in a single end-to-end system. A key component in our network architecure, is that we modulate the input views with per-view, per-scene-depth attention maps that are learned without any supervision. We find that these attention maps indirectly learn a combination of visibility, viewpoint-based weighting, and other factors that when taken into account lead to high-quality view synthesis results. In this section, we discuss the details of our network design (Sec. 5.4.1, 5.4.2, 5.4.3, 5.4.4), data generation (Sec. 5.4.5) and training details (Sec. 5.4.6).

5.4.1 Inputs and Basic Architecture

The inputs to our network are a fixed number of m input images, I_1,I_2,..., I_m from m views under a single directional light. While our architecture can be potentially generalized to setups with other fixed number of views, we consider $m = 6$ in this chapter as discussed in Sec. 5.3. Our network regresses an image I_n and the associated depth map D_n from a novel view point.

This involves aligning and aggregating multi-view inputs, given input view camera parameters $\mathbf{\Theta}_1$, $\mathbf{\Theta}_2$,..., $\mathbf{\Theta}_m$ and novel view camera parameters $\mathbf{\Theta}_n$.

Our network uses a plane sweep volume representation. Plane sweep volumes are constructed by warping colors or features from multiple input views to a novel view at multiple pre-defined depth planes. The warping function, denoted as $\mathcal{W}$, is a homography-based geometric function, and is implemented as a non-learnable (but differentiable) layer. Plane sweep volumes contain geometrically aligned structured data, which is suitable for deep learning-based IBR methods [40]. To help the network better understand the multi-view data, we provide it with information about the input and output viewpoints. Specifically, we supply b_1, b_2,..., b_m, which describe the angular distances from a novel view to each input view, and the depth values d_1, d_2,..., d_p of the p planes in a plane sweep volume, as additional inputs to the network. As such, our network is a regression function $\mathbf{\Phi}$:

$$\begin{aligned} I_n, D_n &= \mathbf{\Phi}(I_1, \mathbf{\Theta}_1, b_1 ..., I_m, \mathbf{\Theta}_m, b_m; d_1, ..., d_p; \mathbf{\Theta}_n) && (5.1) \\ &= \mathbf{\Phi}(\{I_i\}, \{\mathbf{\Theta}_i\}, \{b_i\}; \{d_k\}; \mathbf{\Theta}_n), && (5.2) \end{aligned}$$

where $\{\cdot\}$ represents a set containing either multi-view data (denoted with subscript i) or multiple depths (denoted with subscript k). We use this notation convention in the rest of this chapter.

As shown in Fig. 5.3, our network uses two separate branches. The first, **Corr-Branch**, seeks to analyze multi-view correspondences and reconstruct scene geometry. The second branch, denoted as **Shade-Branch**, reconstructs the output image. Both the correspondence predictor and shading predictor networks use a 3D U-Net architecture, comprised of 3D convolutions, upsampling and downsampling operations, to process their corresponding plane sweep volumes. This allows the network to reason about multi-view scene geometry and appearance while predicting the output view and depth.

The plane sweep volume representation contains incorrect or redundant features (e.g., features that are at incorrect depths or occluded or view-dependent). Therefore, we estimate "attention" maps that account for how much information should be used from each view at each depth, and incorporate factors such as per-view visibility and view weighting. Because these attention maps are likely to be highly correlated with the scene geometry, we predict them jointly with the depth probabilities in Corr-Branch. We pre-modulate the input image pixel volume with these attention maps before it is processed by the shading predictor. This ensures that each depth plane of the input pixel volume only has meaningful color information at the beginning of the shading prediction, leading to signicantly improved view synthesis results.

We provide ground truth rendered images as supervision, which allows the network to reason about scene appearance and photo consistency. We also supervise the output depth images, which allows the network to learn about scene geometry and correspondences. While the learnable parameters of the correspondence and shading branches are separate, their data flows are highly correlated because they share information coming from the image and depth supervision (Fig. 5.3).

5.4.2 Learning Multi-view Correspondences: Corr-Branch

Multi-view reconstruction and re-rendering methods fundamentally rely on finding correspondences across the input images. We achieve this using deep CNNs that process a plane sweep cost volume with deep-learned filters at multiple scales. Depth and visibility information are highly correlated: depth expresses the distance at which multi-view appearance is consistent, and visibility expresses if single-view appearance is consistent with all other views. Therefore, we propose a novel correspondence estimation network, Corr-Branch, to jointly infer depth and visibility-aware attention maps for a novel view.

Corr-Branch consists of a feature extractor $\mathcal{T}$ and a correspondence predictor $\mathcal{C}$. Photometric images are captured under directional lighting and can have high frequency view-dependent specularities which complicate correspondence reasoning. Therefore, we apply a small U-Net style feature extractor, $\mathcal{T}$, to pre-filter each input image I_i. The details of our feature extractor is shown in Fig. 5.4. $\mathcal{T}$ learns to extract specular-invariant features like edges and

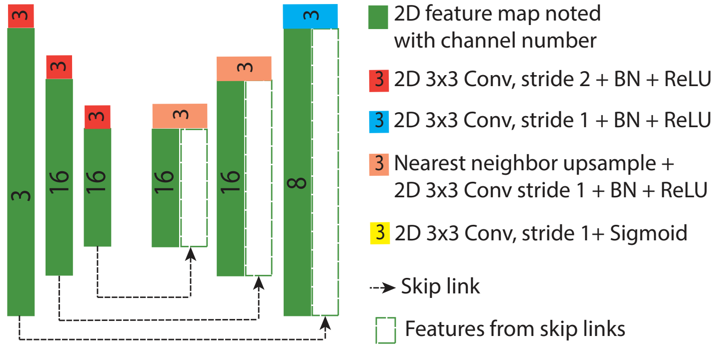

3
3
3
16
3
16
3
16
3
16
3
8
2D feature map noted with channel number
3 2D 3x3 Conv, stride 2 + BN + ReLU
3 2D 3x3 Conv, stride 1 + BN + ReLU
3 Nearest neighbor upsample + 2D 3x3 Conv stride 1 + BN + ReLU
3 2D 3x3 Conv, stride 1+ Sigmoid
Skip link
Features from skip links

Figure 5.4. The details of feature extractor $\mathcal{T}$.

orientations (examples shown in Fig. 5.3) that are meaningful for correspondence estimation. Specifically, $\mathcal{T}$ transforms each 3-channel RGB image I_i to an 8-channel feature map:

$$M_i = \mathcal{T}(I_i). \tag{5.3}$$

Given p discrete, pre-defined depth values $\{d_k | k = 1, .., p\}$, we construct a plane sweep volume $\boldsymbol{M}$ at a novel view, $\boldsymbol{\Theta}_n$, from the extracted per-input-view feature maps $\{M_i | i = 1, .., m\}$:

$$\boldsymbol{M} = \mathcal{W}(\{M_i\}, \{\boldsymbol{\Theta}_i\}; \{d_k\}; \boldsymbol{\Theta}_n). \tag{5.4}$$

$\mathcal{W}$ geometrically warps every M_i onto every depth plane at a distance d_k using input view, $\boldsymbol{\Theta}_i$, and novel view, $\boldsymbol{\Theta}_n$, to form the volume $\boldsymbol{M}$. We use $\boldsymbol{M}_{k,i}$ to denote the warped M_i at the k^{th} depth in $\boldsymbol{M}$. Correspondence inference requires the network to understand photometric consistency across multiple views. We achieve this by processing the volume $\boldsymbol{M}$ with 3D filters with gradually expanding receptive fields. To make it easier for the network to consider multi-view consistency (and inconsistency), we pre-process the feature volume by removing the average multi-view feature at each depth plane as:

$$\tilde{\boldsymbol{M}}_{k,i} = \left(\boldsymbol{M}_{k,i} - \frac{\sum_j \boldsymbol{M}_{k,j}}{m}\right)^2. \tag{5.5}$$

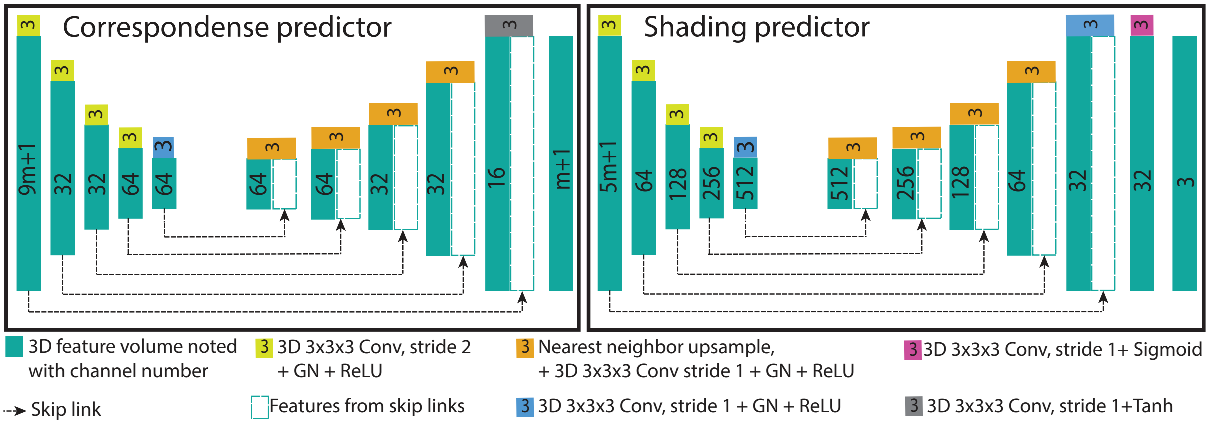

Correspondense predictor
9m+1
32
32
64
64
64
64
32
32
16
m+1
Shading predictor
5m+1
64
128
256
512
512
256
128
64
32
32
3
3D feature volume noted with channel number
3 3D 3x3x3 Conv, stride 2 + GN + ReLU
3 Nearest neighbor upsample, + 3D 3x3x3 Conv stride 1 + GN + ReLU
3 3D 3x3x3 Conv, stride 1+ Sigmoid
Skip link
Features from skip links
3 3D 3x3x3 Conv, stride 1 + GN + ReLU
3 3D 3x3x3 Conv, stride 1+Tanh

Figure 5.5. The details of correspondence predictor $\mathcal{C}$ and shading predictor $\mathcal{S}$.

This operation is similar to variance volumes that have been used in other deep learning-based reconstructions methods [177]. However, while a variance volume expresses only global information across all views, $\tilde{\boldsymbol{M}}$ only subtracts the mean and retains per-view information in the form of per-view maps $\tilde{\boldsymbol{M}}_{k,i}$. This allows our network to infer view-dependent attention maps while also leveraging global information. In fact, if required, the network can compute the variance volume in subsequent convolutional layers by averaging $\tilde{\boldsymbol{M}}$ across views.

$\tilde{\boldsymbol{M}}$ is thus a 3D (depth $\times$ image height $\times$ image width) volume, where each "voxel" has $8m$ channels. We augment these features with view differences b_i and per-plane depth values d_k to make the network utilize the novel view's location (vis-a-vis the input views) for correspondence reasoning:

$$\boldsymbol{N}_k = \tilde{\boldsymbol{M}}_k \oplus \{b_i\} \oplus d_k, \tag{5.6}$$

where $\oplus$ is a per-voxel/per-pixel concatenation operator. Since our views lie on a sphere in our acquisition configuration, we use the cosine of the angle between a pair of views as b_i. The volume, $\boldsymbol{N}$, thus has $9m+1$ channels.

Our correspondence predictor $\mathcal{C}$ is a 3D U-Net style network. It processes volume $\boldsymbol{N}$ through a series of 3D convolutional layers, each followed by group normalization (GN) and ReLU layers. We use downsampling and upsampling along the depth and image dimensions to analyze multi-view correspondence at multiple spatial scales. The details of $\mathcal{C}$ are shown in

Fig. 5.5. The correspondence predictor outputs an $(m+1)$-channel volume, in which the first m channels represent a per-view attention volume $\boldsymbol{V}$ and the last one channel represents a depth probability volume $\boldsymbol{P}$:

$$\boldsymbol{V}, \boldsymbol{P} = \mathcal{C}(\boldsymbol{N}). \tag{5.7}$$

Each channel in $\boldsymbol{V}$ corresponds to the attention information for the corresponding input view, and incorporates both visibility and viewpoint-based weighting information. $\boldsymbol{V}_{k,i}$ is an attention map for the i^{th} view at the k^{th} depth plane; it provides a pixel-wise attention mask that is used during shading prediction. $\boldsymbol{P}_k$, on the other hand, provides a pixel-wise depth probability for the k^{th} depth plane.

$\boldsymbol{P}$ is processed with a depth-wise softmax operation to produce actual probability maps for each depth plane:

$$\boldsymbol{P}^d = \text{soft-max}(\alpha^d \boldsymbol{P}), \tag{5.8}$$

where α^d is a learnable scalar parameter. The output view depth image is finally predicted as:

$$D_n = \sum_k d_k P_k^d. \tag{5.9}$$

We provide ground truth depth images for D_n as supervision. We expect the attention maps and depth estimates to be highly correlated. Therefore, we estimate them from the single correspondence predictor network and share information until the last layer. This ensures that the attention prediction utilizes the depth supervision to learn meaningful features. As noted before, both the attention, $\boldsymbol{V}$, and depth probability, $\boldsymbol{P}$, are provided to the shading prediction branch to help aggregate multi-view appearance.

5.4.3 Learning to Predict Shading: Shade-Branch

Inferring scene appearance from a novel view is a highly challenging task in our configuration: our wide angular baseline input views can see different parts of a scene with limited overlap.

Morevover, complex shading effects like high frequency specular highlights vary significantly across views. We resolve these challenges using our image prediction branch, Shade-Branch. As shown in Fig. 5.3, Shade-Branch has a shading predictor $\mathcal{S}$ that reasons about scene appearance from multi-view images $\{I_i\}$, using the multi-view correspondence information (attention maps, $\boldsymbol{V}$, and depth probability, $\boldsymbol{P}$) predicted by Corr-Branch.

Similar to Eqn. (5.4), a plane sweep volume $\boldsymbol{I}$ is constructed from original images $\{I_i\}$ using homography-based warping $\mathcal{W}$. This volume contains the original color information warped onto multiple planes. This volume can have highly redundant and potentially inconsistent information from multiple views due to strong occlusions. A key feature of our network is that we pre-mask this color volume $\boldsymbol{I}$ using the visibility-aware attention maps inferred from the correspondence branch, to disentangle this redundancy and inconsistency. The masking is achieved by a voxel-wise multiplication, for every view at every depth:

$$\tilde{\boldsymbol{I}}_{k,i} = \boldsymbol{I}_{k,i}\boldsymbol{V}_{k,i}. \tag{5.10}$$

This directly connects Corr-Branch and Shade-Branch; Corr-Branch can thus leverage appearance information from Shade-Branch to estimate attention maps that allow the shading prediction to be as accurate as possible.

Our shading predictor $\mathcal{S}$ is a 3D-Unet style network similar to the correspondence predictor $\mathcal{C}$, but with more channels at each layer for better appearance reasoning. The details of $\mathcal{S}$ are shown in Fig. 5.5. It processes the masked volume $\tilde{\boldsymbol{I}}$, the original attention maps $\boldsymbol{V}$ and other information, and predicts a 3-channel appearance volume

$$\boldsymbol{A} = \mathcal{S}(\tilde{\boldsymbol{I}}, \boldsymbol{V}, \{b_i\}, \{d_k\}), \tag{5.11}$$

where $\tilde{\boldsymbol{I}}$, $\boldsymbol{V}$, $\{b_i\}$, $\{d_k\}$ are concatenated voxel-wise, similar to Eqn. (5.6). Supplying the original attention maps, $\boldsymbol{V}$ (in addition to the modulated appearance volume) gives the network more freedom to reconstruct scene appearance. Note that each plane $\boldsymbol{A}_k$ is a predicted image, containing the predicted appearance of the scene at the k^{th} depth plane. These per-plane predicted images are weighted by the depth probability maps from Corr-Branch to reconstruct the final image as:

$$\tilde{\boldsymbol{A}}_k = \boldsymbol{A}_k P^a_k, \tag{5.12}$$

$$I_n = \sum_k \tilde{\boldsymbol{A}}_k. \tag{5.13}$$

where P^a is the normalized depth probability volume using soft-max and a scalar parameter $\alpha^a = 4$ similar to Eqn. (5.8). Each plane ($\tilde{\boldsymbol{A}}_k$) of $\tilde{\boldsymbol{A}}$, as shown in Fig. 5.3, is a clean weighted image, with depth-incorrect outlier pixels completely masked out.

We provide ground truth rendered novel view images as supervision for I_n, and train our network end to end. As noted before, by transferring information between Shade-Branch and Corr-Branch (in the form of the attention maps and depth estimates), our novel network design consolidates both appearance synthesis and correspondence estimation, allowing the two branches to leverage each other for better inference.

5.4.4 Extension to Multi-light Inputs.

Many photometric applications, like image-based relighting and photometric stereo, acquire photometric images under multiple light sources from a single viewpoint. We propose an extension of our approach to multi-view and *multi-light* datasets. As noted in Sec. 5.3, for this application we capture images under six fixed views and six fixed lights.

Multi-light data allows for better geometric reconstruction by giving us more information about scene appearance [29]. For example, specularities under one light source can disappear under another light source, and shadowed regions under one light can be illuminated by another. This leads to better estimation of normals, edges and other features. We design an extended feature extractor $\mathcal{T}_q$ to take advantage of this for better multi-view correspondence inference. The difference between the single-light $\mathcal{T}$ and the multi-light $\mathcal{T}_q$ feature extractors is merely the number of input channels. $\mathcal{T}_q$ uses a fixed number of structured inputs; the subscript q specifies the number of lights. For each view, we stack q images I_i^1, I_i^2,..,I_i^q under q different light sources together as a $3q$-channel feature map as an input for $\mathcal{T}_q$. While Corr-Branch predicts the attention maps and depth probability maps from multi-light data, the input for our Shade-Branch remains the same; it still predicts a novel view image under a single light source using the multi-view images under that light.

We use our network to synthesize novel views for each input light. In Section 5.6, we show that it can be combined with image-based relighting methods to re-render scene appearance under novel viewpoint and lighting — the full scene acquisition scenario.

5.4.5 Data Generation

To the best of our knowledge, there is no existing large-scale dataset that contains multi-view photometric images. Previous novel view synthesis methods have trained with data from on-line videos [183], car driving scenes [50] or simple objects from specific classes [18], none of which apply to our high-quality appearance acquisition scenario. Therefore, we create a novel large-scale synthetic dataset. As in Xu et al. [172], we procedurally generate shapes by combining primitives with randomly generated bump maps and randomly merging 1 to 9 primitives. We create 1000 training scenes and 50 testing scenes using this method. We texture map these shapes with SVBRDFs from the Adobe Stock 3D material dataset[1]. This dataset contains 1329 SVBRDFs, and we separate it into 1129 training and 200 test materials.

We render our dataset using path tracing with 1000 samples; we use Optix to achieve fast path tracing, similar to [91]. This physically based rendering method ensures that our images contain realistic light transport. We render 512×512-resolution images and tone-map them using gamma 2.2. We render our scenes using the camera configuration described in Sec. 5.3. For each scene, we randomly select a field-of-view angle from 5° to 60°, and correspondingly calculate a distance for cameras based on the angle and the scene's size to ensure good pixel

[1] https://stock.adobe.com/3d-assets

coverage. Each training image set consists of 6 input views and one novel view under a single directional light. For each training scene, we create 30 such sets by randomly placing 30 novel views between the 6 input views and selecting 30 random directional lights for rendering. In total, we have 30000 such sets in our training set. For each test scene, we randomly select 36 novel views and 3 directional lights, and render each view under each light, which creates 108 image sets, for a total of 5400 image sets in our test set.

5.4.6 Training Details

Loss functions. We use an L1 loss on both the ground truth images and depths from each novel view. Specifically, let $\mathscr{L}_a$ be the image L1 loss and $\mathscr{L}_d$ be the depth L1 loss. Our final loss $\mathscr{L}$ is given by

$$\mathscr{L} = \mathscr{L}_a + \beta \mathscr{L}_d. \tag{5.14}$$

We use $\beta = 0.1$ for all our experiments.

Parameters and strategies. Our six views are radially symmetric (see Fig. 5.2); we leverage this symmetry to provide structured inputs to our network and make training easier. Given a novel view, we first rotate the image and depth around the central viewing direction to make its up direction point towards the central input view. We then reorder the input views to achieve a canonical input view layout, where the first view is the central view, and the other views are ordered in counter-clockwise order starting with the view on the left (see Fig. 5.2 right).

We assume that the test scenes are captured by a calibrated spherical gantry and that the physical size of a scene and the distance from a camera to the center of the scene is approximately known. This allows the depth values $\{d_k | k = 1, 2..., p\}$ to be specified correspondingly. While our network is fully convolutional and can support an arbitrary number of depth planes, p, we use $p = 64$ for our training and all experiments. During training, we have perfectly symmetric cameras distributed at a known distance, $\hat{d}$, from the center of a scene. We also know the ground truth scene size Δ (the maximum difference between $\hat{d}$ and each pixel depth). To get $\{d_k\}$, we set $d_1 = \hat{d} - \gamma\Delta$, $d_p = \hat{d} + \gamma\Delta$, and uniformly divide this range for other d_k, which can be expressed by:

$$d_k = \hat{d} - \gamma\Delta + \gamma\frac{2\Delta(k-1)}{p-1}. \quad (5.15)$$

During training, we randomly pick γ between 0.8 to 2.0, so that our network sees different sweep volume scales; we thus only need to specify a roughly correct distance $\hat{d}$ and size Δ at test time for real scenes (using $\gamma = 1$).

We train our networks using three or four NVIDIA Titan Xp or 1080Ti GPUs, using a batch size of 4 for each GPU, for a total batch size of 12 or 16. We apply group normalizations in both $\mathcal{C}$ and $\mathcal{S}$, and observe better performance than batch normalization with our small per-GPU batch size, similar to [168]. During training, we randomly crop 64×64 patches from novel view images and depths for data augmentation. Since our images can have large regions of black background, we only select crops that have at least 50% non-background pixels. Our network generally converges after training for 400 epochs (about 4 days with 4 GPUs).

5.5 Experiments

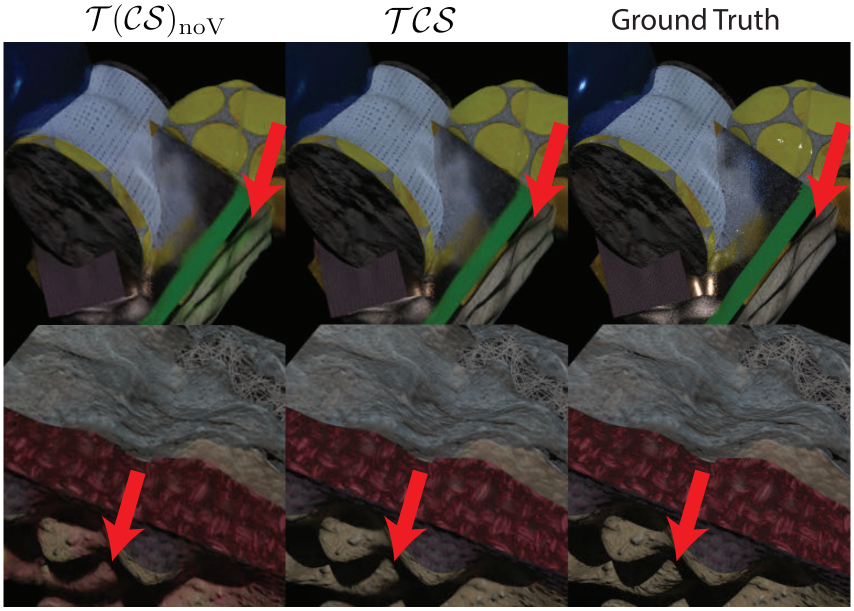
We now present a comprehensive evaluation of our method on both synthetic and real data.

Ablation study on synthetic data. We first justify the design of our network architecture through ablations on the synthetic dataset. We compare our proposed single-light photometric novel view synthesis network, $\mathcal{TCS}$, i.e., with a feature extractor ($\mathcal{T}$) and correspondence and shading predictors ($\mathcal{C}$ and $\mathcal{S}$), against a number of variants. We also compare it to our multi-light network, $\mathcal{T}_6\mathcal{CS}$, with six different directional lights as inputs for Corr-Branch and a multi-light feature extractor. Specifically, we compare against the following networks: $\mathcal{TC}_{\text{noD}}\mathcal{S}$, that doesn't have depth supervision, and $\mathcal{T}(\mathcal{CS})_{\text{noV}}$, a network that does not use attention maps. We evaluate all these networks on our synthetic testing dataset and compare the image L1 loss, PSNR, SSIM,

Table 5.1. Ablation study. We evaluate different versions of our networks on our synthetic testing dataset, and compare the image L1 error, PSNR, SSIM, and Depth L1 error on the central 256×256 crops.

	Image L1	Image PSNR	Image SSIM	Depth L1
$\mathcal{T}(\mathcal{CS})_{\text{noV}}$	0.0451	30.74	0.9391	0.0567
$\mathcal{TC}_{\text{noD}}\mathcal{S}$	0.0345	32.05	0.9520	0.1448
$\mathcal{TCS}$	0.0318	32.61	0.9573	0.0437
$\mathcal{T}_6\mathcal{CS}$	0.0307	33.03	0.9602	0.0246

Figure 5.6. Qualitative comparisons on synthetic test set between $\mathcal{TCS}$ and $\mathcal{T}(\mathcal{CS})_{\text{noV}}$ (i.e., with and without attention maps, respectively). $\mathcal{T}(\mathcal{CS})_{\text{noV}}$ suffers from color bleeding artifacts (red arrows), that are resolved by $\mathcal{TCS}$.



and depth L1 loss. To avoid biases in these metrics from the large black backgrounds in our rendered images, we crop the central 256×256 regions from all testing images for evaluation; these crops have 75% non-background pixels on average. We also calculate depth L1 only for the foreground as we do for training.

The numerical comparisons of these different networks are shown in Tab. 5.1. As demonstrated by $\mathcal{TCS}$ vs $\mathcal{T}(\mathcal{CS})_{\text{noV}}$, our visibility-aware attention maps significantly improve reconstruction performance; image L1 loss reduces by about 30% and is accompanied by a large improvement in PSNR and SSIM. Figure 5.6 shows qualitative comparisons between $\mathcal{TCS}$ and

$\mathcal{T}(\mathcal{CS})_{\mathrm{noV}}$. We observe many color-bleeding artifacts with $\mathcal{TCS}$ because it is unable to resolve the large occlusions in these scenes. We also observe that the attention maps can be inferred and help the synthesis in an unsupervised way; network $\mathcal{TC}_{\mathrm{noD}}\mathcal{S}$ with attention maps, but without depth supervision, performs much better at view synthesis than $\mathcal{T}(\mathcal{CS})_{\mathrm{noV}}$ without attention maps and with depth supervision. These comparisons demonstrate the key role our visibility-aware attention maps play in effectively eliminating the incorrect, inconsistent information in a plane sweep volume. Also, comparing $\mathcal{TC}_{\mathrm{noD}}\mathcal{S}$ vs $\mathcal{TCS}$ shows that depth supervision improves reconstruction accuracy, though in a subtle way.

Finally, when multi-light data is provided, our multi-light network $\mathcal{T}_6\mathcal{CS}$ has the best performance. This confirms that the multi-light feature extractor $\mathcal{T}_6$ extracts better features leading to both better depth prediction and better image synthesis than our single light version. While the multi-light version requires more acquired images, it can be naturally combined with an image-based relighting method to enable view and lighting changes (see Fig. 5.11).

Real data capture. We capture our real scenes—composed of one or more real objects placed on a platform—using a spherical gantry. We capture each scene from the six input views under either a central directional light (for single-light results) or six directional lights (for multi-light results and additional applications). We also capture 50 novel views under these lights, as ground truth to validate our view synthesis quality. Our cameras are about 50cm away from the platform. We thus set $\hat{d} = 50$cm for all our real scenes. While the sizes and scales of our real scenes vary, we find that $\Delta = 6$cm works well for most cases. Our network is robust to these variations because of the randomized Δ during training. We mask out the background of the captured images before passing them to our network. Our network is fully convolutional and we directly apply our method on these images, although we are training on 64×64 crops. We use an Nvidia Titan Xp to process our real results and it takes about 2 seconds to generate a 400×400 image.

Comparisons against previous view synthesis methods. We now compare our method

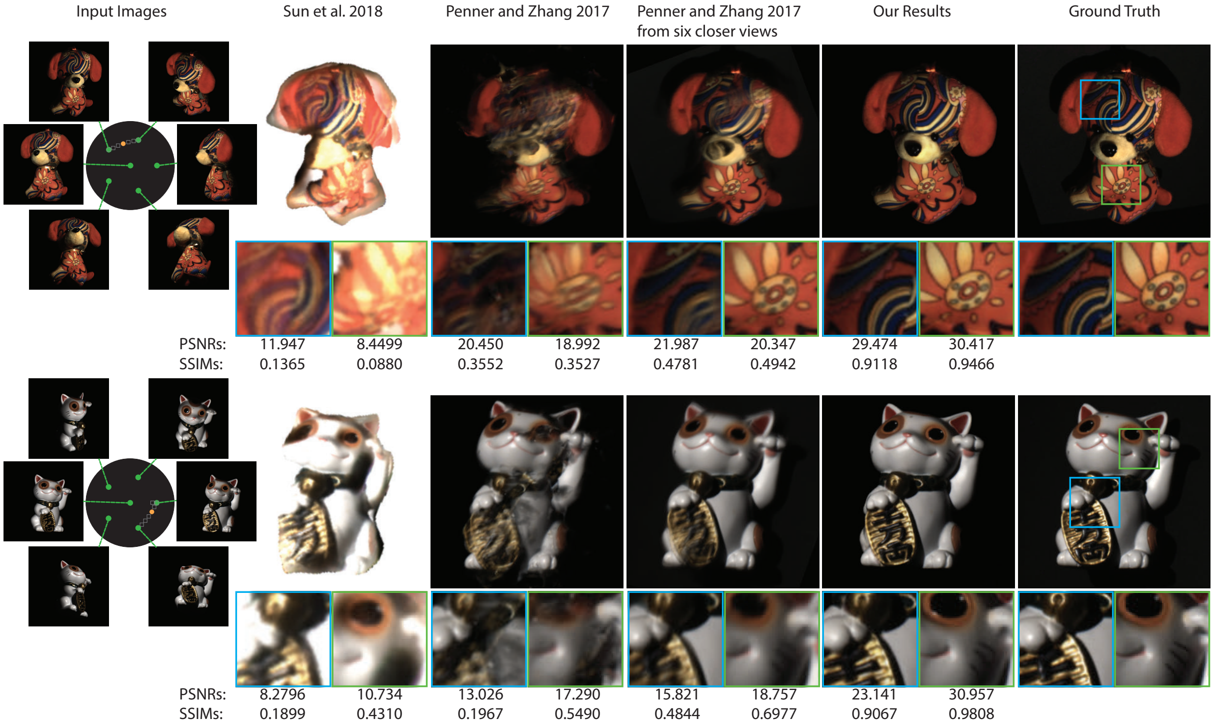

Input Images
Sun et al. 2018
Penner and Zhang 2017
Penner and Zhang 2017 from six closer views
Our Results
Ground Truth
PSNRs: 11.947 8.4499 20.450 18.992 21.987 20.347 29.474 30.417
SSIMs: 0.1365 0.0880 0.3552 0.3527 0.4781 0.4942 0.9118 0.9466
PSNRs: 8.2796 10.734 13.026 17.290 15.821 18.757 23.141 30.957
SSIMs: 0.1899 0.4310 0.1967 0.5490 0.4844 0.6977 0.9067 0.9808

Figure 5.7. Comparisons with flow-based view synthesis [141] (second column) and Soft3D [117] (third column) using wide-baseline inputs (first column, input viewing directions shown in green, and novel view direction in yellow). We also compare with [117] (fourth column) that uses a much closer set of views as input (marked in grey rectangles in the first column). Our results are significantly better than other methods and differences almost imperceptible from ground truth. We show cropped insets of all results with corresponding PSNRs and SSIMs (bottom).

to previous state-of-the-art view synthesis methods. We considered comparing against learning-based IBR methods. However, these methods are usually designed for general IBR applications with densely sampled views [40, 58]. We tried training an implementation of Flynn et al. [40] on our dataset, but it failed to predict reasonable results in our wide-baseline case. Hedman et al. [58] require estimating geometry using multi-view stereo, which does not work from our sparse inputs. Besides, methods designed for specific scenarios, like light fields [72, 140] or pair-view extrapolation [183] are also not easy to apply in our case. We also tried training the released network of [72] on our dataset; it doesn't converge to any reasonable results because of a significantly more challenging task. Therefore, we compare against [117], a state-of-the-art non-learning based IBR method. Note that, Penner and Zhang [117] have already demonstrated that their method performs better than [40] and [72].

We also compare against a flow-based view synthesis method [141]. Directly applying their model, trained on KITTI [50] or ShapeNet [18], doesn’t work on our data. Therefore, we retrain their model using our dataset, albeit with a white background; this is the same scenario Sun et al.[141] use in their paper.

In Fig. 5.7, we show qualitative and quantitative comparisons on two real captured examples. The first scene has a complicated surface texture and the second has complex specularities and hard shadows. In the second and third columns of Fig. 5.7, we compare with [141] and [117] using the same six-view images we use for our method. We can see that our method performs significantly better than the two methods qualitatively, as shown by the synthesized images (details in insets), and quantitatively, as shown by the PSNR and SSIM values. Both previous methods fail to handle this challenging wide-baseline configuration. Sun et al. [141] produce blurred results with no appearance details and many mis-aligned ghosting artifacts. The results of [117] also contain serious ghosting artifacts. Our method, on the other hand, produces photorealistic view synthesis results with significantly higher PSNR and SSIM values. As shown in the insets, our method recovers both the complicated texture of the first example as well as the challenging hard shadows and view-dependent specularities in the second example.

We also select six closer views as input for [117]. As shown in the fourth column, these "easier" inputs improve their results. However, their results with the small-baseline inputs are still worse than our results from the six wide-baseline inputs, highlighting the accuracy and robustness of our method.

View synthesis on real photometric data. Figures 5.1 and 5.8 show our view synthesis results from our single-light network compared with captured ground truth. Our method produces photo-realistic novel view images for these real scenes, which accurately match the ground truth. As demonstrated in many examples, our method generates high-quality view interpolation results

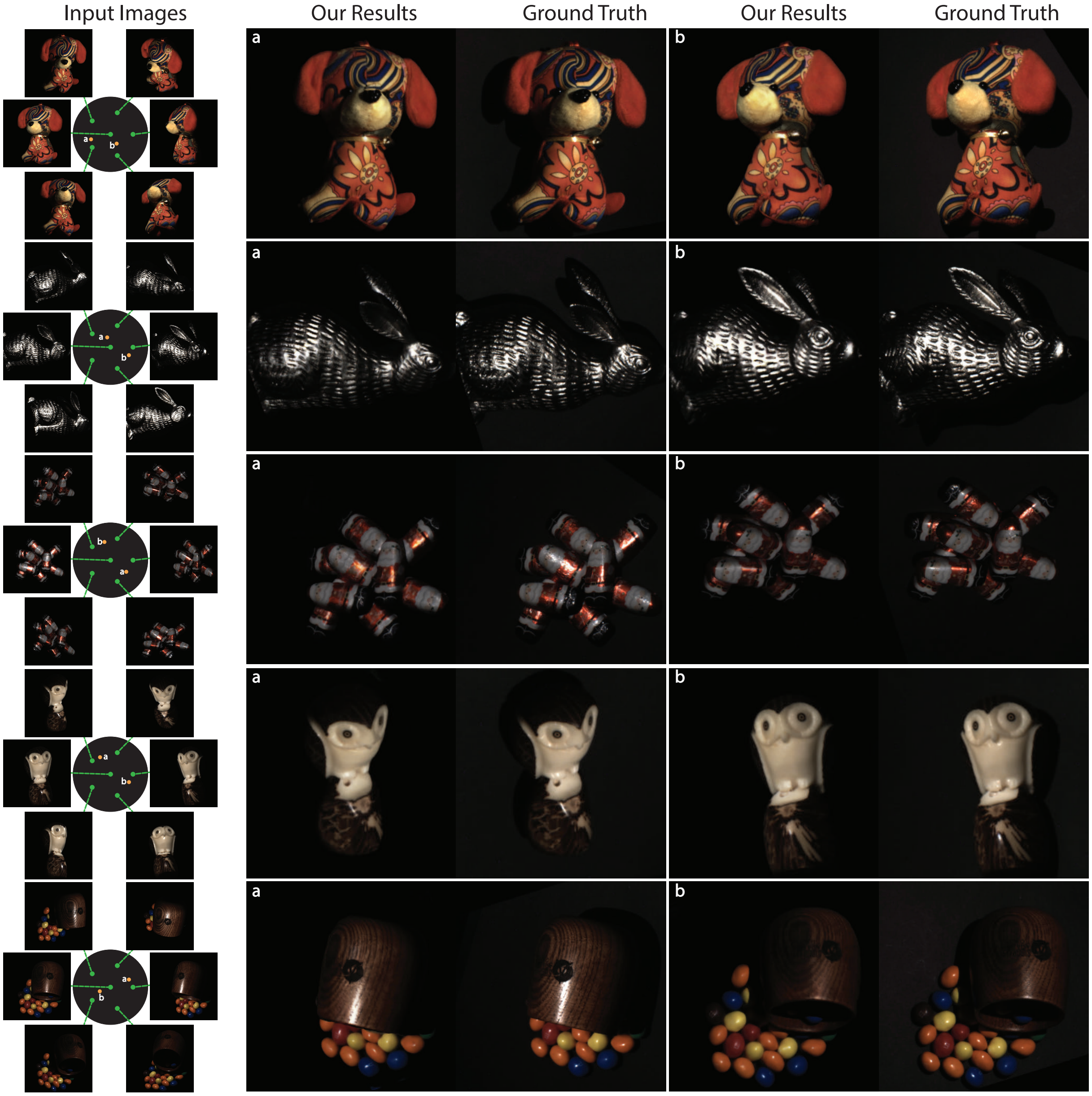


Figure 5.8. Novel view synthesis results from our single-light network on real scenes. For each scene, we show two novel view synthesis results (second and forth columns) compared with captured ground truth images (third and fifth columns), whose viewing directions are marked in yellow with corresponding labels (a and b). We also show the inputs in the first column marked with corresponding viewing directions in green.

even at challenging viewing directions that are close to the boundary of the pentagonal cone, where very limited input information can be used. These results are consistent across a wide variety of scenes, in terms of both materials (pottery, cloth, mental, wood, plastic and candy) and

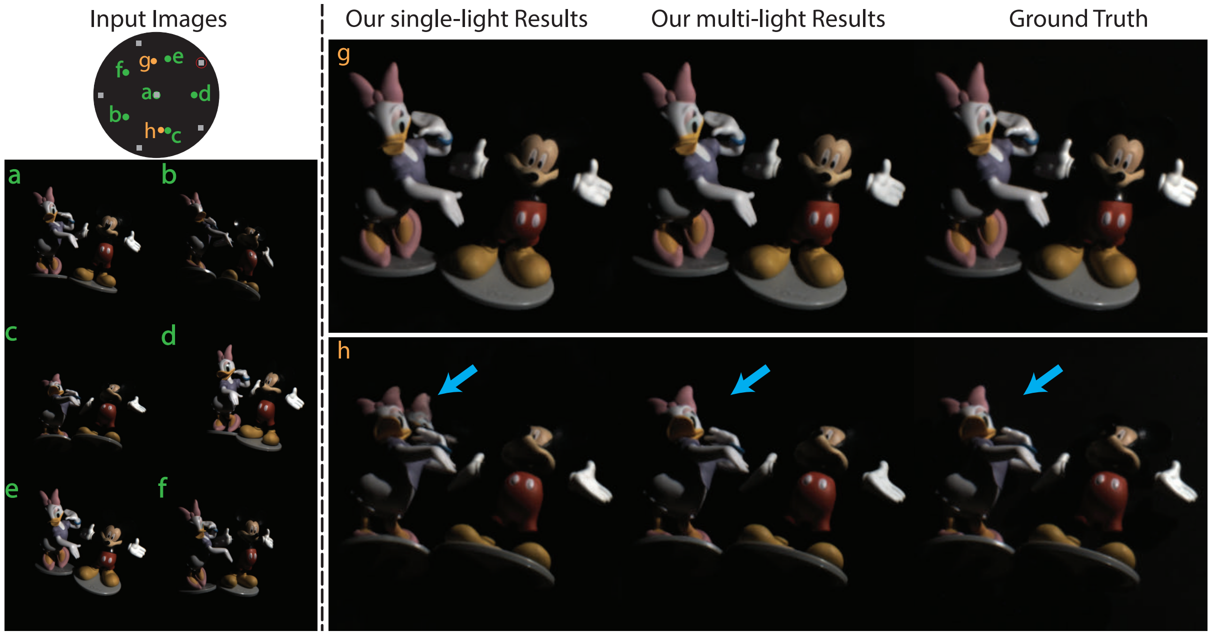


Figure 5.9. Single-light vs. multi-light network comparison. For a complex scene captured under a challenging light direction (marked by red hollow circle), our single-light network may generate ghosting artifacts from some challenging viewing directions (second row, marked by blue arrows). Our multi-light network resolves these issues using images under multiple light sources (marked by gray). For most viewing directions, our single-light network produces high-quality results (first row). The six input images are shown on the left, with light directions marked in green with labels (a-f), and novel views marked in yellow with labels (g,h).

geometry (single and multiple objects; small and big objects).

Comparison between single-light and multi-light networks. We show a challenging scene under directional lighting in Fig. 5.9, and compare our single-light network with our extended multi-light network. The scene contains thin structures (like the arms and thumbs) which are very distinct from our training geometry primitives. These highly non-convex structures exhibit challenging cast shadows that complicate correspondence inference. Nevertheless, our single-light network performs well for most viewing directions (e.g. Fig. 5.9.g). As shown in Fig. 5.9.h, our single-light network may generate obvious ghosting artifacts for some challenging directions, but our multi-light network can resolve these issues thanks to more reliable correspondence inference from multi-light images.

Limitations. Our method only handles opaque scenes, which is a limitation of our

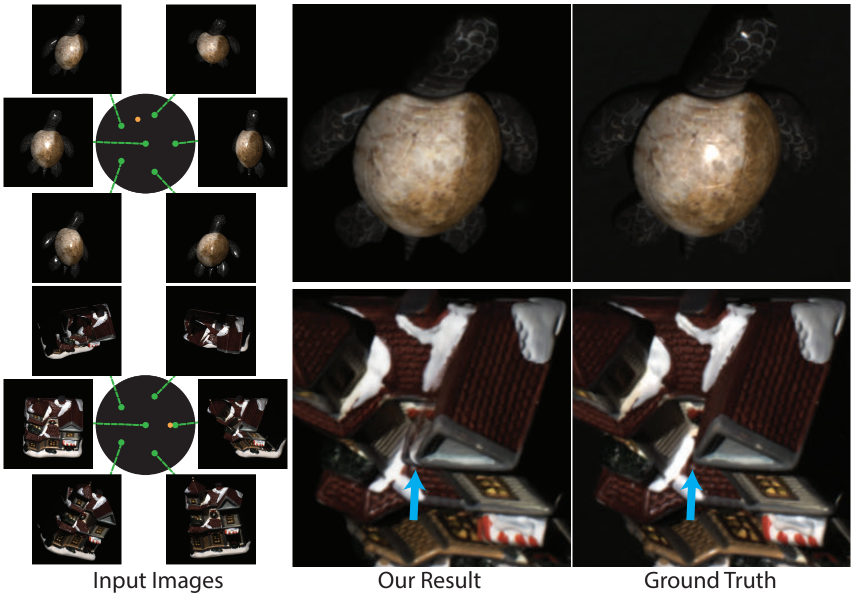


Figure 5.10. Limitations. Our method fails to reconstruct sharp specularies that have long-range motion (top) and highly non-convex occlusions (bottom).

training dataset. Our network is trained on 64 × 64 cropped images, which limits the spatial scale of appearance reasoning. Consequently, long-range effects like sharp specularities that move significantly are not reconstructed well (see Fig. 5.10). Our method might blur sharp specularities (see Fig. 5.8). Also, our network generates blurred results with ghosting for highly non-convex scenes with parts that are visible in only one or two views (see Fig. 5.10).

5.6 Additional Applications

Our view synthesis method can be combined with other scene acquisition and rendering techniques to enable a broad set of applications. We now demonstrate a few examples.

Novel view relighting Our method can synthesize novel views from images captured under different directional lights. These, in turn, can be used with image-based relighting methods to enable rendering under novel view and lighting. One such relighting example is shown in Fig. 5.1b3. We apply our multi-light network with six different directional lights and synthesize novel view images for each light separately (see Fig. 5.11 top). We train the

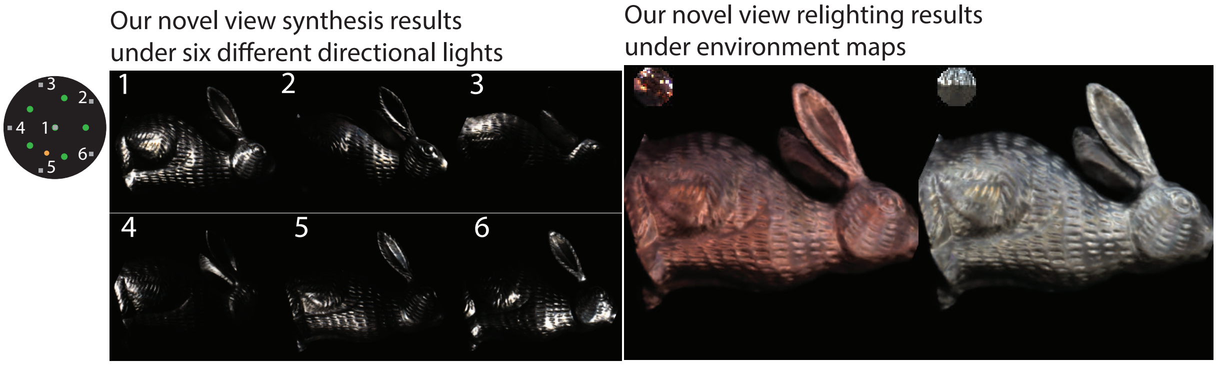

Our novel view synthesis results
under six different directional lights
Our novel view relighting results
under environment maps
1
2
3
4
5
6
1
2
3
4
5
6

Figure 5.11. Novel view relighting using our novel view synthesis results. We apply our multi-light network to synthesize six novel view images under six directional lights (marked in gray with labels 1-6), each synthesized from the same six views (marked in green). We use these high-quality synthesis results as inputs for the deep image-based relighting technique in Chapter 4, and create relighting results (on the bottom) under environment maps (shown on the top left of each result).

Relight-Net network in Chapter 4 for our six-light setting. We pass the synthesized novel view images for the six lights to this network to generate images under novel directional lights. Similar to Chapter 4, we achieve relighting under novel environment maps by linearly combining the relit images as shown in Fig. 5.11 bottom. Note that our network and the Relight-Net are trained separately, without end-to-end refinement or any other special processing.

While our results from a novel view under changing environment map often look realistic, some aspects still need improvment. For example, some blurriness, incorrect shadow motion and temporal inconsistency that become obvious when changing the view under a novel environment map. Jointly training our network with the Relight-Net using a larger task-specific training dataset can potentially resolve, or at least alleviate, these issues. That said, to our knowledge, this is the first attempt at synthesizing a full reflectance field, enabling changes to both lighting and viewpoint, from such sparse samples (36 images from 6 views under 6 lights).

Multi-view stereo Multi-view stereo methods often require a dense set of input views, and can fail to reconstruct complete hole-free geometry for sparse views such as ours. We apply our method to "densify" the captured scene and synthesize 56 novel view images around a scene from six images. We pass these 56 synthesized images to a multi-view stereo system COLMAP

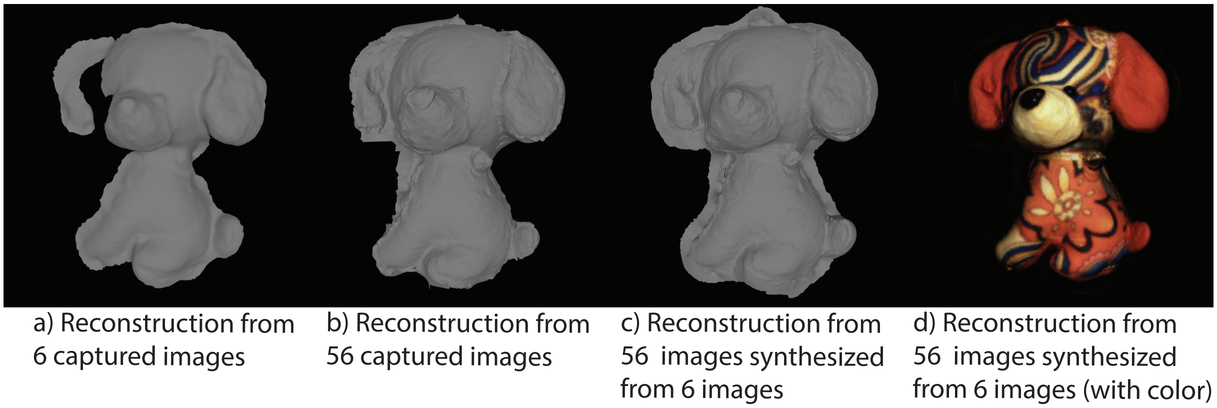

a) Reconstruction from 6 captured images
b) Reconstruction from 56 captured images
c) Reconstruction from 56 images synthesized from 6 images
d) Reconstruction from 56 images synthesized from 6 images (with color)

Figure 5.12. Multi-view stereo reconstruction from our synthesized images. We synthesize 56 high-quality novel views from six images and use them as inputs for a multi-view stereo algorithm, COLMAP [131], to generate 3D reconstructions (c,d) that are comparable to a reconstruction from 56 captured images (b). COLMAP reconstructs incomplete geometry with holes from only the six sparse images (a).

[131] and achieve 3D reconstruction of the scene (see Fig. 5.12 c and d). As a baseline, we pass the 56 captured ground truth images to COLMAP for reconstruction and we observe qualitatively similar results as shown in Fig. 5.12. Note that COLMAP reconstructs incomplete geometry with missing parts from the original six sparse views. By making MVS methods work better with such sparse viewpoints, our method makes them more robust and general.

5.7 Summary

We have demonstrated a method to synthesize photometric scene appearance at a wide range of novel viewpoints from a sparse set of only six images captured at large baselines. This is in contrast to previous methods that rely on densely sampling the scene with hundreds of viewpoints. We achieve this by training a novel deep CNN that can simultaneously infer correspondences and shading from structured photometric images. Our network predicts visibility-aware attention maps that effectively address photometric and geometric inconsistencies and allow for the accurate aggregation of multi-view scene appearance. We present evaluations and comparisons to previous view synthesis methods, and show that we can generate significantly more accurate and photorealistic images across a wide range of scenes. Fundamentally, our work takes a step towards capturing and rendering scene appearance from sparse image sets. This is a classic problem in vision and graphics and we believe that our work can enable many other applications. For example, we demonstrate that our synthesized images can be used to achieve novel view relighting and multi-view stereo from sparse images. In the future, it would be interesting to explore extensions of our technique to other challenging scene acquisition tasks, like multi-view BRDF reconstruction, 360° scene reconstruction, and the acquisition of dynamic scene appearance from sparse images.

This chapter is a reformatted version of the material as it appears in “Deep View Synthesis from Sparse Photometric Images,” Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, Ravi Ramamoorthi, ACM Transactions on Graphics (TOG) 38 (4), 2019 [170]. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Conclusion and Future Work

In this dissertation, we mainly focus on advancing the acquisition and reconstruction efficiency for appearance acquisition of real scenes. In Chapter 2, we demonstrate that the uncertainty in deep learning-based depth prediction can be properly incorporated to refine the reconstruction. We present a novel UCS-Net that leverages differentiable uncertainty estimates to adaptively distribute sparse spatially varying depth hypotheses in the scene space. Our UCS-Net achieves high-quality geometry reconstruction with high accuracy and completeness, while consuming very low computation and memory costs. In Chapter 3, we present a comprehensive error analysis for a PCA based BRDF reconstruction framework, which allows us to search the BRDF sampling space and find the optimal light-view samples with minimal reconstruction error. We demonstrate that, using the optimal sampling directions, high-fidelity BRDFs can be reconstructed from only two near-field images. Our work significantly reduces the required number of samples in geometry reconstruction (reducing the number of depth hypotheses) and material reflectance acquisition (reducing the number of light-view samples).

We have also presented image-based appearance acquisition techniques that bypass explicit geometry and reflectance reconstruction, which implicitly model the scene components for photo-realistic image synthesis under new conditions. In Chapter 4, we introduce a novel deep neural network for image-based relighting, which simultaneously learns optimized input lighting directions and a corresponding CNN based relighting function. We show that our relighting network can relight a scene realistically with complex shadows and specularities from only five images under five optimized directional lights. In Chapter 5, we propose a deep neural network for view synthesis, which leverages 3D CNNs to process plane sweep volumes to jointly learn the multi-view correspondence and appearance. Our network leverages visibility-aware attention to effectively aggregate the appearance from sparse multi-view images. We demonstrate that our approach can synthesize high-quality novel-view images with realistic specularities and occlusions from only six widely-spaced input views. Our work significantly reduces the required number of samples in image-based relighting (reducing the number of lights) and view synthesis (reducing the number of views).

We have presented approaches that advance the state-of-the-art in various appearance acquisition problems, including both explicit scene reconstruction and image-based acquisition. We believe the future of appearance acquisition is full of possibilities. We now briefly discuss some promising future directions.

Natural illumination. While our geometry reconstruction (Chapter 2) is done under natural illumination, our reflectance (Chapter 3) and image-base appearance acquisition techniques (Chapters 4, 5) still require controlled lighting. Extensions of these to natural illumination would further advance the practicality. Our recent works have relaxed the lighting constraint by applying a semi-controlled lighting (a dominant known flashlight plus unknown dim background illumination) [91] or restricting the object category (human portrait) [142]. In general, high-quality appearance acquisition under natural illumination for general scenes would require the geometry and reflectance modeling (explicitly or implicitly) done jointly with lighting estimation, to disentangle the highly ambiguous scene appearance.

Multi-view reconstruction. We have demonstrated that high-quality scene geometry can be reconstructed from multi-view images using deep learning techniques (see Chapter 2). In Chapter 5 (Sec. 5.6 and Fig. 5.12), we also show that geometry reconstruction can be potentially facilitated by combining with realistic view synthesis techniques. In the future, we believe view synthesis techniques and deep stereo techniques can be further combined to achieve complete scene reconstruction of geometry and materials from sparse inputs. In a consequent work [11], we explore this multi-view reconstruction problem with structured inputs under controlled lighting. It will be highly interesting to see if high-quality multi-view reconstruction can be done with unstructured inputs under even natural illumination for very high practicality.

Singl-image appearance acquistion. We have demonstrated a sequence of approaches that can acquire the appearance of real scenes with only a few images (see Chapters 3, 4, 5). To further improve the acquisition efficiency and practicality, single-image appearance acquisition is a very intriguing direction; this requires strong prior knowledge, which is hard to be modeled traditionally, but can be now effectively learned by deep neural networks. In recent works, we exploit single-view reconstruction in [91], leveraging data priors learned from synthetic scenes similar to Chapters 4, 5; we also explore techniques for more realistic single-image portrait relighting in [142] using stronger category-specific data priors learned from real portrait data. Future single-image acquisition for arbitrary shape and reflectance can be potentially improved by novel high-quality fully-labeled real datasets and advanced image generation techniques like GANs [55].

Dynamic appearance acquistion. This thesis mainly exploits appearance acquisition problems for static scenes. It is very interesting, though very challenging, to extend some of our techniques to dynamic scenes, which is a more practical acquisition problem, since the human is dynamically interacting with the world very actively. Previously, we have also studied dynamic hair capture in [173] with densely sampled input views and temporal frames; video relighting for human performance has been achieved with controlled high-speed capture setups [159, 36]. Recently, deep learning based dynamic capture has also been demonstrated [94], which still leverages dense inputs with controlled setups. It would be very interesting to see if these dynamic techniques can be combined with our sparse sampling techniques to achieve dynamic appearance acquisition from sparse inputs.

Appendix A

Details of Uncertainty Estimation.

We discuss additional analysis and details about our uncertainty estimation evaluated on the DTU validate set. We have shown the average lengths of the uncertainty intervals and the corresponding average sampling distances between the depth planes of the ATVs in Tab. 2.5 in Chapter 2. We now show the histograms of the uncertainty interval length in Fig. A.1 to better illustrate the distributions of the interval length. We mark the average lengths and the median lengths in the histograms.

Note that, the distributions of the two ATVs are unimodal, in which most lengths distribute around the peaks; however, the average interval lengths differ much from the modes in the histograms, because of small portions of the intervals that have very large uncertainty. This means that using the average interval lengths – as what we do for Tab. 2.5 – to discuss the depth-wise sampling is in fact underestimating the sampling efficiency we have achieved for most of the pixels, though our average lengths are good and correspond to a high sampling rate. Therefore, we additionally show the median values in the histograms, which are less sensitive to the large-value outliers and are more representative than the mean values for these distributions. As shown in Fig. A.1, the median interval lengths of the two ATVs are 12.01mm and 2.71mm respectively, which are closer to the peaks of the histograms; these lengths correspond to depth-wise sampling distances of 0.38mm and 0.34mm, given our specified 32 and 8 depth planes. These are significantly higher sampling rates than previous works, such as MVSNet [177] –

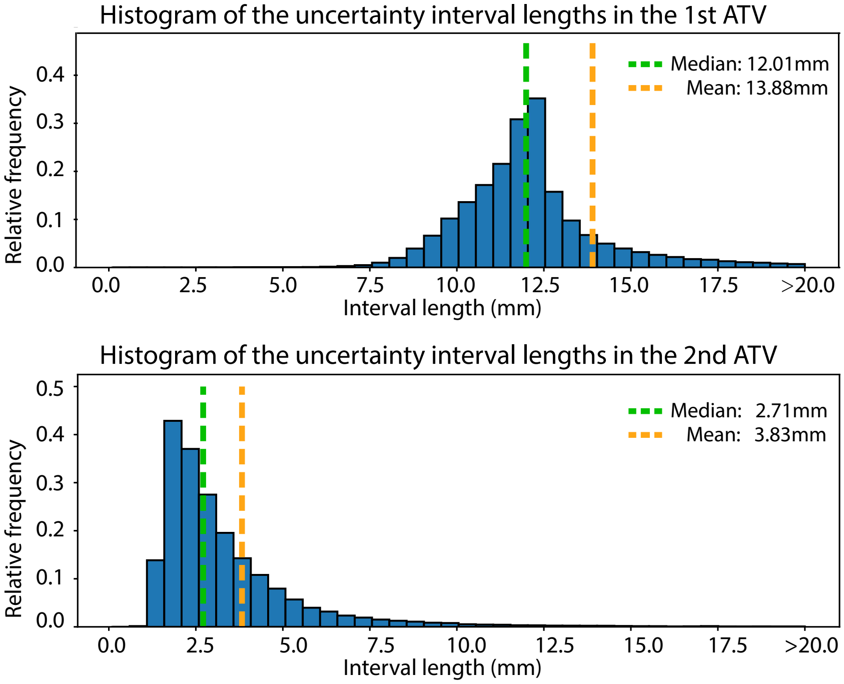


Figure A.1. Histograms of the uncertainty interval lengths. We create bins for every 0.5mm to compute the histograms of the lengths of the uncertainty intervals in the two ATVs. We mark the median and the mean values of the lengths in the histograms.

which uses 256 planes to obtain a sampling distance of 1.99mm – and RMVSNet [178] – which uses 512 planes to obtain a sampling distance of 0.99mm. Our ATV allows for highly efficient spatial partitioning, which achieves a high sampling rate with a small number of depth planes.

Appendix B

Details of Reconstruction Error (E_{recon} in Eqn. 3.12)

We analyze Eqn. 3.12 in more detail, also relating it to the condition number metric. It is convenient to denote $y = Qc$, where c is the accurate coefficient vector to reconstruct the BRDF. In this case, noting $Q^{+}Q = I$,

$$E_{\text{recon}} = \left| Q\left(Q^{+} - (SQ)^{+}_{\eta} S\right) Qc \right| = \left| Q\left(I - (SQ)^{+}_{\eta} SQ\right) c \right|. \tag{B.1}$$

Now, let us denote the SVD of $SQ = \tilde{Q}$ as $A\Lambda B^{T}$. From equation 3.5, $(SQ)^{+}_{\eta} = \tilde{Q}^{+}_{\eta} = B\Lambda^{+}_{\eta}A^{T}$. Now,

$$(SQ)^{+}_{\eta}(SQ) = B\Lambda^{+}_{\eta}A^{T}A\Lambda B^{T} = B\Lambda' B^{T}, \tag{B.2}$$

where $\Lambda' = \Lambda^{+}_{\eta}\Lambda$ is a diagonal matrix. If the singular values in Λ are σ (and those in Λ^{+}_{η} are $\sigma/(\sigma^{2}+\eta)$), then the singular values in Λ' are $\sigma^{2}/(\sigma^{2}+\eta)$. Further simplifying,

$$E_{\text{recon}} = \left| Q\left(I - B\Lambda' B^{T}\right) c \right| = \left| Q\left(B\Gamma B^{T}\right) c \right|, \tag{B.3}$$

where Γ is also a diagonal matrix with singular values $1 - \frac{\sigma^{2}}{\sigma^{2}+\eta} = \eta/(\sigma^{2}+\eta)$. To understand E_{recon}, we care about the singular values in Γ. The largest singular value is given by the minimum σ_{min}, with value $\eta/(\sigma^{2}_{\text{min}}+\eta)$. In general, we will reduce E_{recon} if we avoid small σ. Indeed, the condition number optimization affects the σ values and tries to make σ_{min} larger to reduce the condition number. However, it is not explicitly minimizing the above expression. In contrast, our approach explicitly considers the end-to-end system, as well as the effect of Q, the MERL BRDF materials encoded in the coefficient vector c, and the full spectrum of singular values, to fully minimize the error E_{recon}.

Appendix C

Point-Sampled BRDF Measurement.

Chapter 3 discusses near-field image-based BRDF measurement. Here, we show that the new error metric also somewhat improves point-sampled BRDF acquisition.

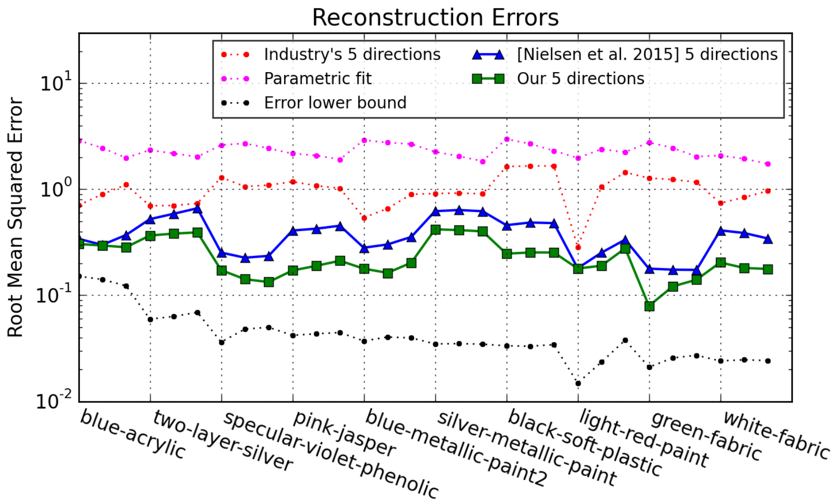


Figure C.1. Comparison of reconstruction with our new optimized 5 directions, and those from [109], parametric fits, and industry-standard directions. Our method (green curve) produces lower error than previous work (blue curve) on each BRDF.

We compare our results to [109] with 5 directions in Fig. C.1. (The dotted black curve at the bottom is the lower bound when using all of the input directions, essentially the unavoidable error $E_{\text{deviation}}$.) Note that this evaluation is identical to Fig. 8 in their paper, using the same graphs for their method, as well as parametric fits and the industry-standard 5 directions in [160]. It is clear that we have somewhat lower error. This is not surprising since these results are

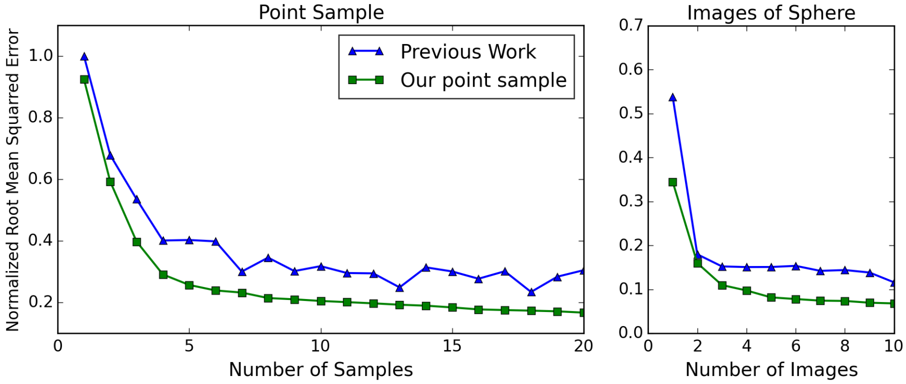


Figure C.2. Reconstruction error versus number of measurements. We obtain a smooth graph, strictly lower error than previous work.

computed assuming the observations are accurate without noise, while the condition number metric measures only sensitivity to noise, not reconstruction error. The supplementary material shows similar results for an example with 20 measurements and 2% noise. Nevertheless, minimizing the condition number is a reasonable heuristic for this setup.

We can also plot the average error over the unknown samples in the MERL BRDF database vs. the number of measurements *n* in Fig. C.2. For both standard point-sampled acquisition, and the image-based spherical acquisition method of [98] (extended to use optimal directions computed with either our error metric or using condition number), our method gives somewhat lower error. Another important observation is the shape of the curves. The result from [109] oscillates somewhat, since the condition number metric is not directly tied to (or always monotonic with) the actual error. By minimizing the actual expected reconstruction error, we obtain a smooth graph. The supplementary material provides our improved point-sampling directions, and comparisons for a few materials from the MERL database. In some cases we do qualitatively better, while there is a minor improvement in other cases. In general, our 5 directions is comparable to 20 samples using the previous condition number metric.

Appendix D

Optimal Lights for Additional Configurations.

Chapter 4 shows several learned lighting distributions for different k in Fig. 4.6. We now show more optimal directions we learn from our joint training process in Fig. D.1.

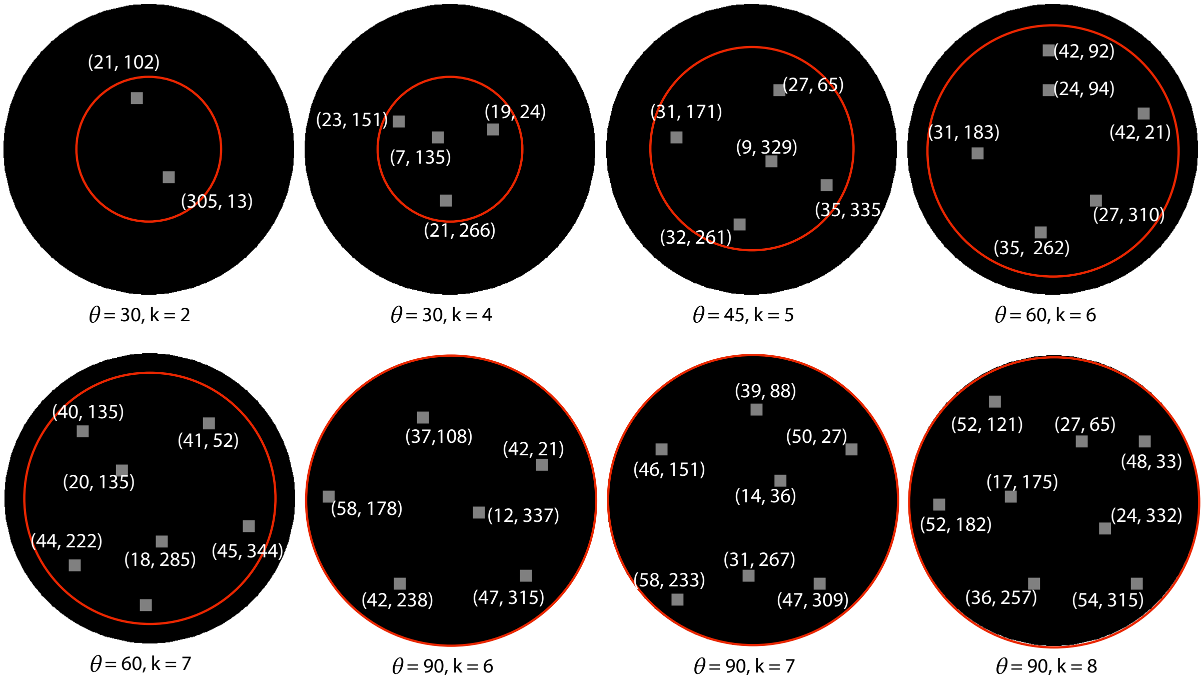


Figure D.1. Lighting configurations that are not shown in Fig. 4.6. We represent directions using the standard (θ, ϕ) spherical parameterization.

We also show a comparison between a Relight-Net trained with $\theta = 90$, $k = 5$ and one with $\theta = 45$, $k = 4$ in Fig. D.2. We can see that for the same relighting direction, a network

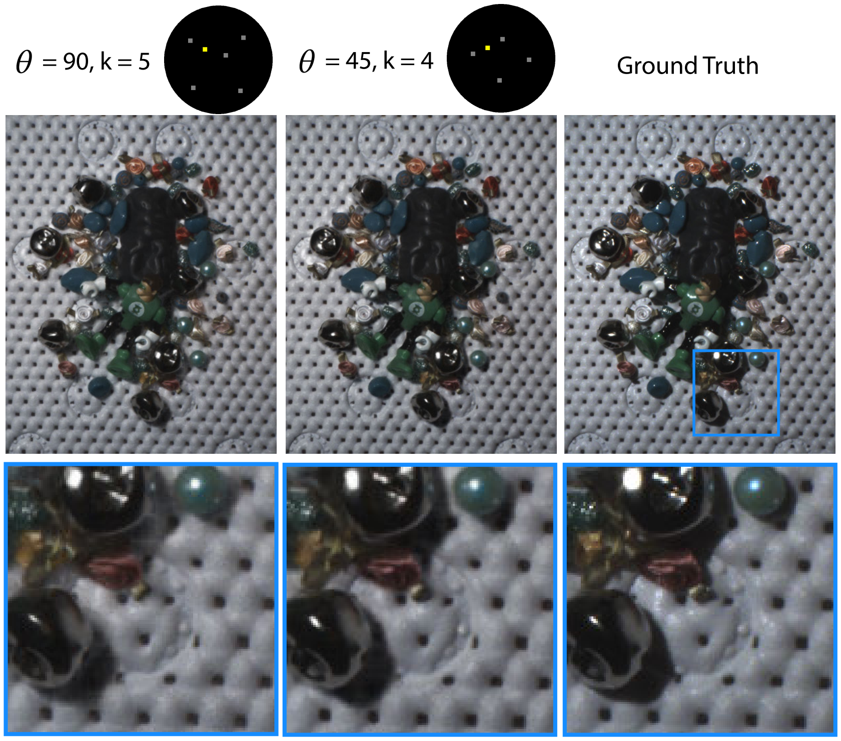


Figure D.2. Comparison between $\theta = 90$, $k = 5$ and $\theta = 45$, $k = 4$. When our network is trained on a smaller cone, a smaller number of samples are required to achieve equal or better performance.

trained on a smaller cone achieves the same and even better results with fewer samples. This is compatible with what the PSNR distributions show in Fig. 4.8 in Chapter 4. Thus, if we only seek to do relighting within a smaller cone of directions, it is possible to use fewer input images, and a network specifically trained for a smaller angular cone θ.

Bibliography

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016.

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *ICML*, pages 40–49, 2018.

[3] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Practical SVBRDF capture in the frequency domain. *ACM Transactions on Graphics (TOG)*, 32(4):110, 2013.

[4] Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. Two-shot SVBRDF capture for stationary materials. *ACM Transactions on Graphics (TOG)*, 34(4), 2015.

[5] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2D-3D model alignment via surface normal prediction. *CVPR*, 2016.

[6] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence*, 37(8):1670–1687, 2015.

[7] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):218–233, February 2003.

[8] Konstantinos Batsos, Changjiang Cai, and Philippos Mordohai. Cbmv: A coalesced bidirectional matching volume for disparity estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2060–2069, 2018.

[9] Peter N. Belhumeur and David J. Kriegman. What is the set of images of an object under all possible illumination conditions? *International Journal of Computer Vision*, 28(3):245–260, Jul 1998.

[10] Sai Bi, Nima Khademi Kalantari, and Ravi Ramamoorthi. Patch-based optimization for image-based texture mapping. *ACM Trans. Graph.*, 36(4):106–1, 2017.

[11] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. *arXiv preprint arXiv:2003.12642*, 2020.

[12] A. Brady, J. Lawrence, P. Peers, and W. Weimer. genBRDF: discovering new analytic BRDFs with genetic programming. *ACM Transactions on Graphics (TOG)*, 33(4), 2014.

[13] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *SIGGRAPH*, pages 425–432. ACM, 2001.

[14] Brett Burley. Physically-based shading at disney. In *ACM SIGGRAPH 2012 Courses*, 2012.

[15] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conference on Computer Vision*, pages 766–779. Springer, 2008.

[16] Ayan Chakrabarti. Learning sensor multiplexing design through back-propagation. In *Advances in Neural Information Processing Systems*, pages 3081–3089, 2016.

[17] Manmohan Chandraker. The information available to a moving observer on shape with unknown, isotropic brdfs. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1283–1297, 2016.

[18] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[19] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 32(3):30, 2013.

[20] Gaurav Chaurasia, Olga Sorkine, and George Drettakis. Silhouette-aware warping for image-based rendering. In *Computer Graphics Forum*, volume 30, pages 1223–1232. Wiley Online Library, 2011.

[21] Anpei Chen, Minye Wu, Yingliang Zhang, Nianyi Li, Jie Lu, Shenghua Gao, and Jingyi Yu. Deep surface light fields. *Proc. ACM Comput. Graph. Interact. Tech.*, 1(1):14:1–14:17, July 2018.

[22] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *ICCV*, 2019.

[23] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of SIGGRAPH*, pages 279–288, 1993.

[24] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *arXiv preprint arXiv:1812.02822*, 2018.

[25] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. *arXiv preprint arXiv:1911.12012*, 2019.

[26] Łukasz Dąbała, Matthias Ziegler, Piotr Didyk, Frederik Zilly, Joachim Keinert, Karol Myszkowski, H-P Seidel, Przemyslaw Rokita, and Tobias Ritschel. Efficient multi-image correspondences for on-line light field video processing. In *Computer Graphics Forum*, volume 35, pages 401–410. Wiley Online Library, 2016.

[27] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5868–5877, 2017.

[28] Kristin J. Dana and Jing Wang. Device for convenient measurement of spatially varying bidirectional reflectance. *J. Opt. Soc. Am. A*, 21(1):1–12, 2004.

[29] James Davis, Diego Nehab, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Spacetime Stereo: A unifying framework for depth from triangulation. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 27(2):296–302, February 2005.

[30] Jeremy S De Bonet and Paul Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 418–425, 1999.

[31] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156. ACM Press/Addison-Wesley Publishing Co., 2000.

[32] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20. ACM, 1996.

[33] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image SVBRDF capture with a rendering-aware deep network. *ACM Transactions on Graphics*, 37(4):128, 2018.

[34] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015.

[35] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *ICCV*, 2015.

[36] Per Einarsson, Charles-Felix Chabert, Andrew Jones, Wan-Chun Ma, Bruce Lamond, Tim Hawkins, Mark T Bolas, Sebastian Sylwan, and Paul E Debevec. Relighting human locomotion with flowed reflectance fields. *Rendering techniques*, 2006:17th, 2006.

[37] Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson De Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. In *Computer graphics forum*, volume 27, pages 409–418. Wiley Online Library, 2008.

[38] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004.

[39] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[40] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016.

[41] Sing Choong Foo. *A gonioreflectometer for measuring the bidirectional reflectance of material for use in illumination computation*. PhD thesis, Citeseer, 1997.

[42] Martin Fuchs, Volker Blanz, Hendrik Lensch, and Hans-Peter Seidel. Adaptive sampling of reflectance fields. *ACM Transactions on Graphics (TOG)*, 26(2):10, 2007.

[43] Martin Fuchs, Volker Blanz, Hendrik P.A. Lensch, and Hans-Peter Seidel. Adaptive sampling of reflectance fields. *ACM Transactions on Graphics (TOG)*, 26(2):10, 2007.

[44] Ryo Furukawa, Hiroshi Kawasaki, Katsushi Ikeuchi, and Masao Sakauchi. Appearance based object modeling using texture database: Acquisition compression and rendering. In *Rendering Techniques*, pages 257–266, 2002.

[45] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010.

[46] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015.

[47] Silvano Galliani and Konrad Schindler. Just look at the image: viewpoint-specific surface normal prediction for improved multi-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5479–5487, 2016.

[48] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 9(4), 2017.

[49] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[50] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361. IEEE, 2012.

[51] Stamatios Georgoulis, Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Tinne Tuytelaars, and Luc Van Gool. What is around the camera? In *ICCV*, 2017.

[52] Abhijeet Ghosh, Shruthi Achutha, Wolfgang Heidrich, and Matthew O'Toole. BRDF acquisition with basis illumination. In *International Conference on Computer Vision*, pages 1–8, 2007.

[53] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[54] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, 2010.

[55] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[56] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1996.

[57] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *ICCV*, pages 1586–1594, 2017.

[58] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. In *SIGGRAPH Asia 2018 Technical Papers*, page 257. ACM, 2018.

[59] Paul Henderson and Vittorio Ferrari. Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision*, pages 1–20, 2019.

[60] Aaron Hertzmann and Steven M Seitz. Shape and materials by example: A photometric stereo approach. In *Computer Vision and Pattern Recognition*, volume 1, pages 533–540. IEEE, 2003.

[61] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *CVPR*, 2017.

[62] Michael Holroyd, Jason Lawrence, and Todd Zickler. A coaxial optical scanner for synchronous acquisition of 3D geometry and surface reflectance. *ACM Transactions on Graphics*, 29(4):99, 2010.

[63] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *CVPR*, pages 2821–2830, 2018.

[64] Z. Hui and A. C. Sankaranarayanan. Shape and spatially-varying reflectance estimation from virtual exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(10):2060–2073, Oct 2017.

[65] Zhuo Hui, Kalyan Sunkavalli, Joon-Young Lee, Sunil Hadap, Jian Wang, and Aswin C Sankaranarayanan. Reflectance capture using univariate sampling of brdfs. In *ICCV*, pages 5362–5370, 2017.

[66] Richard S Hunter and Deane B Judd. Development of a method of classifying paints according to gloss. *ASTM Bulletin*, (97):11–18, 1939.

[67] Richard Sewall Hunter. *The measurement of appearance*. John Wiley & Sons, 1987.

[68] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In-So Kweon. Dpsnet: End-to-end deep plane sweep stereo. In *7th International Conference on Learning Representations, ICLR 2019*. International Conference on Learning Representations, ICLR, 2019.

[69] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems*, pages 2807–2817, 2018.

[70] Wenzel Jakob. Mitsuba renderer, 2010. http://www.mitsuba-renderer.org.

[71] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. SurfaceNet: An end-to-end 3D neural network for multiview stereopsis. In *ICCV*, pages 2307–2315, 2017.

[72] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):193, 2016.

[73] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7122–7131, 2018.

[74] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018.

[75] Sing Bing Kang, Richard Szeliski, and Jinxiang Chai. Handling occlusions in dense multi-view stereo. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[76] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NIPS*, pages 365–376, 2017.

[77] K. Karner, H. Mayer, and M. Gervautz. An image-based measurement system for anisotropic reflection. *Computer Graphics Forum (EUROGRAPHICS 96)*, 15(3):119–128, 1996.

[78] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018.

[79] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):29, 2013.

[80] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 66–75, 2017.

[81] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017.

[82] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European conference on computer vision*, pages 82–96. Springer, 2002.

[83] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000.

[84] Lubor Ladicky, Olivier Saurer, SoHyeon Jeong, Fabio Maninchedda, and Marc Pollefeys. From point clouds to mesh using regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3893–3902, 2017.

[85] Eric P Lafortune and Yves D Willems. Bi-directional path tracing. 1993.

[86] Hendrik P. A. Lensch, Jochen Lang, Asla M. Sá, and Hans peter Seidel. Planned sampling of spatially varying BRDFs. *Computer Graphics Forum*, 22(3):473–482, 2003.

[87] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[88] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):418–433, 2005.

[89] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 36(4):45, 2017.

[90] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: Svbrdf acquisition with a single mobile phone image. In *ECCV*, 2018.

[91] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. In *SIGGRAPH Asia 2018*, page 269. ACM, 2018.

[92] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[93] Guilin Liu, Duygu Ceylan, Ersin Yumer, Jimei Yang, and Jyh-Ming Lien. Material editing using a physically based rendering network. In *ICCV*, pages 2261–2269, 2017.

[94] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics (TOG)*, 38(4):65, 2019.

[95] J. Low, J. Kronander, A. Ynnerman, and J. Unger. BRDF models for accurate and efficient rendering of glossy surfaces. *ACM Transactions on Graphics (TOG)*, 31(1), 2012.

[96] Dhruv Mahajan, Ira Kemelmacher Shlizerman, Ravi Ramamoorthi, and Peter Belhumeur. A theory of locally low dimensional light transport. *ACM Trans. Graph.*, 26(3), July 2007.

[97] Tom Malzbender, Dan Gelb, and Hans Wolters. Polynomial texture maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 519–528, 2001.

[98] S Marschner, S Westin, E Lafortune, K Torrance, and D Greenberg. Image-Based BRDF Measurement including Human Skin. In *Eurographics Rendering Workshop*, pages 139–152, 2000.

[99] Wojciech Matusik, Matthew Loper, and Hanspeter Pfister. Progressively-Refined Reflectance Functions from Natural Illumination. In Alexander Keller and Henrik Wann Jensen, editors, *Eurographics Workshop on Rendering*, 2004.

[100] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics (TOG)*, 22(3):759–769, 2003.

[101] Wojciech Matusik, Hanspeter Pfister, Matthew Brand, and Leonard McMillan. Efficient isotropic BRDF measurement. In *Eurographics Rendering Workshop*, pages 241–247, 2003.

[102] M McCool, J Ang, and A Ahmad. Homomorphic Factorization of BRDFs for High-Performance Rendering. In *SIGGRAPH 01*, pages 171–178, 2001.

[103] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. *arXiv preprint arXiv:1812.03828*, 2018.

[104] M. Meyer and J. Anderson. Key point subspace acceleration and soft caching. *ACM Transactions on Graphics (TOG)*, 26(3), 2007.

[105] Giljoo Nam, Joo Ho Lee, Diego Gutierrez, and Min H Kim. Practical SVBRDF acquisition of 3D objects with unstructured flash photography. In *SIGGRAPH Asia 2018*, page 267. ACM, 2018.

[106] Shree K. Nayar, Peter N. Belhumeur, and Terry E. Boult. Lighting sensitive display. *ACM Trans. Graph.*, 23(4):963–979, October 2004.

[107] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 376–381. ACM, 2003.

[108] Addy Ngan, Frédo Durand, and Wojciech Matusik. Experimental analysis of BRDF models. In *Eurographics Symposium on Rendering*, pages 117–126, 2005.

[109] J. Boll Nielsen, H. Jensen, and R. Ramamoorthi. On optimal, minimal BRDF sampling for reflectance acquisition. *ACM Transactions on Graphics (TOG)*, 34(6), 2015.

[110] Jannik Boll Nielsen, Henrik Wann Jensen, and Ravi Ramamoorthi. On optimal, minimal brdf sampling for reflectance acquisition. *ACM Transactions on Graphics (TOG)*, 34(6):1–11, 2015.

[111] Tobias Noll, Didier Stricker, Johannes Kohler, and Gerd Reis. A full-spherical device for simultaneous geometry and reflectance acquisition. In *Proceedings of the on Applications of Computer Vision*, WACV, pages 355–362. IEEE Computer Society, 2013.

[112] Geoffrey Oxholm and Ko Nishino. Shape and reflectance estimation in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(2):376–389, 2016.

[113] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 702–711. IEEE, 2017.

[114] Despoina Paschalidou, Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3897–3906, 2018.

[115] Pieter Peers and Philip Dutré. Inferring reflectance functions from wavelet noise. In *Proceedings of the Sixteenth Eurographics conference on Rendering Techniques*, pages 173–182. Eurographics Association, 2005.

[116] Pieter Peers, Dhruv K Mahajan, Bruce Lamond, Abhijeet Ghosh, Wojciech Matusik, Ravi Ramamoorthi, and Paul Debevec. Compressive light transport sensing. *ACM Transactions on Graphics (TOG)*, 28(1):3, 2009.

[117] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):235, 2017.

[118] J-P Pons, Renaud Keriven, O Faugeras, and Gerardo Hermosillo. Variational stereovision and 3d scene flow estimation with statistical similarity measures. In *IEEE 9th International Conference on Computer Vision*, page 597. IEEE, 2003.

[119] Ravi Ramamoorthi and Pat Hanrahan. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *J. Opt. Soc. Am. A*, 18(10):2448–2459, Oct 2001.

[120] Dikpal Reddy, Ravi Ramamoorthi, and Brian Curless. Frequency-space decomposition and acquisition of light transport under spatially varying illumination. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 596–610, Berlin, Heidelberg, 2012. Springer-Verlag.

[121] Konstantinos Rematas, Tobias Ritschel, Mario Fritz, Efstratios Gavves, and Tinne Tuytelaars. Deep reflectance maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4508–4516, 2016.

[122] Peiran Ren, Yue Dong, Stephen Lin, Xin Tong, and Baining Guo. Image based relighting using neural networks. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015.

[123] Peiran Ren, Jiaping Wang, John Snyder, Xin Tong, and Baining Guo. Pocket reflectometry. *ACM Transactions on Graphics*, 30(4), 2011.

[124] Stephan R Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *CVPR*, pages 1936–1944, 2018.

[125] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. OctnetFusion: Learning depth fusion from data. In *2017 International Conference on 3D Vision*, pages 57–66. IEEE, 2017.

[126] Fabiano Romeiro, Yuriy Vasilyev, and Todd Zickler. Passive reflectometry. In *European Conf. Computer Vision*, pages 859–872, 2008.

[127] Fabiano Romeiro and Todd Zickler. Blind reflectometry. In *European Conference on Computer Vision*, pages 45–58, 2010.

[128] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015.

[129] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[130] Szymon Rusinkiewicz. A new change of variables for efficient BRDF representation. In *Eurographics Rendering Workshop*, pages 11–22, 1998.

[131] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016.

[132] Christopher Schwartz, Michael Weinmann, Roland Ruiters, and Reinhard Klein. Integrated high-quality acquisition of geometry and appearance for cultural heritage. In *VAST*, pages 25–32. Eurographics Association, October 2011.

[133] Christopher Schwartz, Michael Weinmann, Roland Ruiters, and Reinhard Klein. Integrated high-quality acquisition of geometry and appearance for cultural heritage. In *VAST*, volume 2011, pages 25–32, 2011.

[134] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

[135] Amnon Shashua. On photometric issues in 3d visual recognition from a single 2d image. *International Journal of Computer Vision*, 21(1):99–122, Jan 1997.

[136] Daeyun Shin, Zhile Ren, Erik B Sudderth, and Charless C Fowlkes. 3d scene reconstruction with multi-layer depth and epipolar transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2172–2182, 2019.

[137] Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6040–6049, 2017.

[138] Sudipta Sinha, Drew Steedly, and Rick Szeliski. Piecewise planar stereo for image-based rendering. 2009.

[139] Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph.*, 22(3):382–391, July 2003.

[140] Pratul P Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. Learning to synthesize a 4d rgbd light field from a single image. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2262–2270, 2017.

[141] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[142] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Transactions on Graphics (Proceedings SIGGRAPH)*, 2019.

[143] Kalyan Sunkavalli, Todd Zickler, and Hanspeter Pfister. Visibility subspaces: Uncalibrated photometric stereo with shadows. In *ECCV*, pages 251–264, 2010.

[144] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018.

[145] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Single-view to multi-view: Reconstructing unseen views with a convolutional network. *CoRR abs/1511.06702*, 1(2):2, 2015.

[146] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012.

[147] Kenneth E Torrance and Ephraim M Sparrow. Theory for off-specular reflection from roughened surfaces. *JOSA*, 57(9):1105–1112, 1967.

[148] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2626–2634, 2017.

[149] Borom Tunwattanapong, Graham Fyffe, Paul Graham, Jay Busch, Xueming Yu, Abhijeet Ghosh, and Paul Debevec. Acquiring reflectance and shape from continuous spherical harmonic illumination. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.

[150] Ali Osman Ulusoy, Andreas Geiger, and Michael J Black. Towards probabilistic volumetric reconstruction using ray potentials. In *2015 International Conference on 3D Vision*, pages 10–18. IEEE, 2015.

[151] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5038–5047, 2017.

[152] Suren Vagharshakyan, Robert Bregovic, and Atanas Gotchev. Light field reconstruction using shearlet transform. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 40(1):133–147, 2018.

[153] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206. Eurographics Association, 2007.

[154] Jiaping Wang, Yue Dong, Xin Tong, Zhouchen Lin, and Baining Guo. Kernel nyström method for light transport. In *ACM Transactions on Graphics (TOG)*, volume 28, page 29. ACM, 2009.

[155] Jinglu Wang, Bo Sun, and Yan Lu. Mvpnet: Multi-view point regression networks for 3d object reconstruction from a single image. *arXiv preprint arXiv:1811.09410*, 2018.

[156] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D mesh models from single rgb images. In *ECCV*, pages 52–67, 2018.

[157] Gregory J Ward. Measuring and modeling anisotropic reflection. In *SIGGRAPH 92*, pages 265–272, 1992.

[158] Michael Weinmann and Reinhard Klein. Advances in geometry and reflectance acquisition (course notes). In *SIGGRAPH Asia 2015 Courses*, pages 1:1–1:71, 2015.

[159] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics (TOG)*, 24(3):756–764, 2005.

[160] Harold B. Westlund and Gary W. Meyer. Applying appearance standards to light reflection models. In *SIGGRAPH 01*, pages 501–510, 2001.

[161] Tim Weyrich, Jason Lawrence, Hendrik P. A. Lensch, Szymon Rusinkiewicz, and Todd Zickler. Principles of appearance acquisition and representation. *Found. Trends. Comput. Graph. Vis.*, 4(2):75–191, February 2009.

[162] Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, and Markus Gross. Analysis of human faces using a measurement-based skin reflectance model. *ACM Trans. Graph.*, 25(3):1013–1024, July 2006.

[163] Daniel N Wood, Daniel I Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 287–296. ACM Press/Addison-Wesley Publishing Co., 2000.

[164] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19:19 – 19 – 6, 1980.

[165] Robert J Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139, 1980.

[166] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *NIPS*, pages 540–550, 2017.

[167] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–662, 2018.

[168] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[169] Rui Xia, Yue Dong, Pieter Peers, and Xin Tong. Recovering shape and spatially-varying surface reflectance under unknown illumination. *ACM Transactions on Graphics*, 35(6):187, 2016.

[170] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics*, 38(4):76, 2019.

[171] Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. Minimal brdf sampling for two-shot near-field reflectance acquisition. *ACM Transactions on Graphics*, 35(6):188, 2016.

[172] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics*, 37(4):126, 2018.

[173] Zexiang Xu, Hsiang-Tao Wu, Lvdi Wang, Changxi Zheng, Xin Tong, and Yue Qi. Dynamic hair capture using spacetime optimization. *ACM Transactions on Graphics (TOG)*, 33(6):224, 2014.

[174] Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems*, pages 1099–1107, 2015.

[175] Li Yao, Yunjian Liu, and Weixin Xu. Real-time virtual view synthesis using light field. *EURASIP Journal on Image and Video Processing*, 2016(1):25, 2016.

[176] Yao Yao, Shiwei Li, Siyu Zhu, Hanyu Deng, Tian Fang, and Long Quan. Relative camera refinement for accurate dense reconstruction. In *2017 International Conference on 3D Vision (3DV)*, pages 185–194. IEEE, 2017.

[177] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *ECCV*, pages 767–783, 2018.

[178] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019.

[179] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. In *Advances in Neural Information Processing Systems*, pages 2263–2274, 2018.

[180] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 822–838, 2018.

[181] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3D reconstruction with consumer depth cameras. *ACM Transactions on Graphics*, 33(4):155, 2014.

[182] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[183] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[184] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision (ECCV)*, pages 286–301. Springer, 2016.

[185] Zhenglong Zhou, Zhe Wu, and Ping Tan. Multi-view photometric stereo with spatially varying isotropic materials. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1482–1489, 2013.

[186] Zhiming Zhou, Guojun Chen, Yue Dong, David Wipf, Yong Yu, John Snyder, and Xin Tong. Sparse-as-possible SVBRDF acquisition. *ACM Transactions on Graphics*, 35(6):189, 2016.